Self-Localization of Ad-Hoc Arrays Using Time Difference of Arrivals

Lin Wang, Tsz-Kin Hon, Joshua D. Reiss, and Andrea Cavallaro

***Abstract*—We investigate the problem of sensor and source joint localization using time-difference of arrivals (TDOAs) of an ad-hoc array. A major challenge is that the TDOAs contain unknown time offsets between asynchronous sensors. To address this problem, we propose a low-rank approximation method that does not need any prior knowledge of sensor and source locations or timing information. At first, we construct a pseudo time of arrival (TOA) matrix by introducing two sets of unknown timing parameters (source onset times and device capture times) into the current TDOA matrix. Then we propose a Gauss-Newton low-rank approximation algorithm to jointly identify the two sets of unknown timing parameters, exploiting the low-rank property embedded in the pseudo TOA matrix. We derive the boundaries of the timing parameters to reduce the initialization space and employ a multi-initialization scheme. Finally, we use the estimated timing parameters to correct the pseudo TOA matrix, which is further applied to sensor and source localization. Experimental results show that the proposed approach outperforms state-of-the-art algorithms.**

***Index Terms*—Ad-hoc array, low rank approximation, self-localization, time-difference of arrival.**

I. Introduction

Ad-hoc sensor networks composed of randomly distributed and independent recording devices, such as smartphones, wireless microphones, and laptops, have been attracting increased interest due to their flexibility in sensor placement [1], [2]. However, the geometrical configuration of an ad-hoc array is generally unknown and may change with time. Device localization is a very important task in this context. Although not necessary in blind source separation [3] and adaptive beamforming [4], a precise knowledge of the device locations is still required in many applications such as fixed beamforming, source localization and tracking [5]. A straightforward approach for device localization is to measure the distance among all the device pairs and to apply closed-form estimators to recover their spatial locations [6]–[9]. The inter-device distance can be directly measured using tapes or laser pointers, which is time consuming [10]. The inter-device distance can also be computed based on the acoustical transfer delays between two devices, by actively communicating calibration sounds between independent devices [11], [12]. Active communication requires a specially designed network interface or software, which is not available in many recording devices. Passive device localization instead exploits external acoustic events emitted from discrete spatial positions. Two types of information, time of arrival (TOA) and time difference of arrival (TDOA), can be estimated from microphone recordings and be used to jointly localize sensors and sources [13], [14]. If the source signal is known beforehand (predefined sound), its TOA for each individual microphone is obtained through cross-correlation with the given signal. If the source signal is unknown (ambient sound), the TDOA for a pair of microphones is estimated through cross-correlation of the two microphone signals.

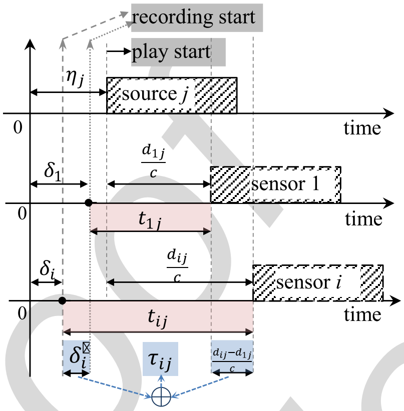


Fig. 1. Illustration of TOA (t_{1j} and t_{ij}) and TDOA (τ_{ij}) for source j and two sensors 1 and i. The source has unknown onset time η_j. The sensors have unknown capture start times δ_1 and δ_i, and unknown time offset δ_i'.

A challenge that arises for an ad-hoc array is that its devices are usually not synchronized. As a result, practical TOA measurements are biased because they may include an unknown source onset time and an unknown device capture time (Fig. 1). Similarly, practical TDOA measurements may include an unknown time offset between a pair of recordings. These time uncertainties make the sensor localization problem harder [14], [17].

Manuscript received May 16, 2015; revised August 21, 2015; revised August 21, 2015; accepted October 11, 2015. This work was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/K007491/1.

L. Wang is with Centre for Intelligent Sensing, Queen Mary University of London, London, E1 4NS, UK (e-mail: lin.wang@qmul.ac.uk; Phone: +44 20 7882 7549).

T.-K. Hon is with Centre for Intelligent Sensing, Queen Mary University of London, London, E1 4NS, UK (e-mail: tsz.kin.hon@qmul.ac.uk, thomas_htk@hotmail.com; Phone: +44 20 7882 7549).

J. D. Reiss is with Centre for Intelligent Sensing, Queen Mary University of London, London, E1 4NS, UK (e-mail: joshua.reiss@qmul.ac.uk; Phone: +44 20 7882 7982).

A. Cavallaro is with Centre for Intelligent Sensing, Queen Mary University of London, London, E1 4NS, UK (e-mail: a.cavallaro@qmul.ac.uk; Phone: +44 20 7882 5165).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2498130

To address the unknown timing information, one can estimate all the unknown parameters, including sensor and source locations and time offsets, simultaneously [5]. The former approach usually leads to the minimization of a non-convex cost function. To overcome local minima problems, some approaches introduce additional constraints, such as co-located sensor-source pairs [14] or far-field sound sources [15]. However, these constraints are not always satisfied in practical applications. Alternatively, one can estimate the unknown timing information, and then estimate the sensor and source locations [19]. This simplifies the problem by decomposing it into two independent stages. The low-rank structure embedded in the TOA measurements can be exploited to estimate the unknown onset times and capture times [16]–[18]. These approaches typically involve a gradient-based optimization procedure, which is sensitive to local minima. Moreover, existing algorithms usually simplify the optimization by assuming either the onset times or the capture times to be known. To the best of our knowledge, a general but effective solution is still missing.

In this paper we present a method that jointly estimates the sensor and source locations from the biased TDOA measurements. We employ the two-stage estimation framework, i.e., timing information estimation followed by sensor and source localization. When estimating the timing information, the existing low-rank approximation techniques are not applicable to TDOA because the latter does not contain the desired low-rank structure as TOA does. To address this challenge, we construct a pseudo TOA matrix by introducing two sets of unknown timing parameters (source onset times and device capture times) into the current TDOA matrix. To estimate these two parameters from the pseudo TOA, we propose a Gauss-Newton low-rank approximation algorithm.

The main novelty of the proposed algorithm is summarized as follows. The proposed complete solution is in contrast to the state-of-the-art timing information estimation algorithms [16]–[18], which assume that only one set of parameters (either onset times or capture times) are unknown. The proposed algorithm has several features which are helpful to tackle local minima problems. First, it is shown that the onset times and capture times in the pseudo TOA matrix can be represented as functions of the sensor and source locations, and particularly, these functions are independent of the TOA data. This feature makes it possible to derive the boundary of the parameter space, and to reduce the initialization space with a reasonable assumption of the maximum sensor-source distance. Second, the algorithm can converge quickly to either local or global minima, and typically shows evident distinction between the two minima. This feature makes it both computationally and technically feasible to employ a multi-initialization scheme to increase the probability of reaching a global minimum.

The paper is organized as follows. Section II overviews related work of self-localization. Section III formulates the problem and introduces existing solutions. The proposed method, including pseudo TDOA matrix construction, timing information estimation, and geometrical information estimation, is described in Sections IV–VI, respectively. Performance evaluation is conducted in Section VII and conclusions are drawn in Section VIII.

II. Related Work

Passive device self-localization approaches can be divided into two classes, namely pairwise distance estimation and joint sensor and source localization (Table I).

Pairwise distance estimation approaches recover the distance between each pair of devices and then the relative locations of all the devices. One can assume a diffuse noise field and estimate the microphone pairwise distance by fitting the measured noise coherence to the theoretical model of the field [20], [21]. Although this solution alleviates the need for activating several sources, it is only feasible for small arrays, and requires the microphones to be synchronized in order to calculate the coherence correctly. Moreover, the diffuse noise assumption is not always satisfied. Another approach is to determine the inter-device distance by estimating the minimum and maximum TDOA values [22]–[25]. By assuming that the minimum and maximum TDOAs come from sources located at the end-fire directions, the inter-device distance can be calculated without knowing the time offset. However, the assumption of end-fire sources is not always satisfied.

Joint localization approaches estimate the locations of sensors and sources simultaneously using acoustic events emitted from various spatial locations. Information that can be used for the joint localization task includes TOA, TDOA, and received signal energy. The TOA measurement may be biased by the unknown source onset time and device capture time, while the TDOA measurement may be biased by the unknown time offset. To minimize the influence of unknown timing information, two schemes can be employed: one-stage estimation [5], [14] and two-stage estimation [16], [17].

The *one-stage* scheme estimates the unknown position and timing information simultaneously via a maximum likelihood estimation, using either TOA or TDOA information [5], [14], [26]. The method presented in [26] jointly estimates the locations and onset times, utilizing the TOAs of calibration signals. The presented method in [14] jointly estimates the locations and capture times from the TOAs or TDOAs of calibration signals. A closed-form solution is also derived by assuming pairs of co-located sensors and sources, e.g., in laptops or tablets [14]. The method presented in [5] jointly estimates the locations and capture times from the TDOAs measured from ambient sounds. An auxiliary function-based iterative algorithm, which shows a better convergence property, is proposed for the optimization procedure. If the timing information is already known, the one-stage estimation can be performed by exploiting the affine geometry of the sensors and sources [13], [15]. A singular value decomposition (SVD) based low-rank matrix factorization approach is employed to reduce the number of unknown parameters, which are further estimated through a much simpler non-linear optimization procedure. The method in [15] estimates the locations of the sensors and sources from the TOAs of the sources, which are assumed to be at the far field and hence have the same onset time. This method is further extended to a general case which does not rely on the far-field assumption [13], [30]. A closed-form solution can be derived by assuming a pair of co-located sensor and source [13]. Besides TDOA and

TABLE I
SUMMARY OF DEVICE SELF-LOCALIZATION ALGORITHMS. (DT: DEVICE CAPTURE TIME; ST: SOURCE START TIME; DL: DEVICE LOCATION; SL: SOURCE LOCATION; PD: PAIRWISE DISTANCE).

Reference	Acoustic Signal	Information Used	Approach		Constraint	Parameters Needed	Parameters Estimated
[20], [21]	ambient sound	mic signal coherence	pairwise distance + close-form estimator	noise coherence match	diffuse noise field, small array	DT	PD
[22]–[25]	ambient sound	TDOA		Min/Max TDOA estimation	end-fire sources	–	PD
[5]	ambient sound	TDOA	joint sensor/source localization	one-stage estimation	–	–	DT, DL, SL
[14]					co-located sensor/source	–	DT, DL, SL
[15]					far-field sources	DT	DL, SL
[13]	calibration signal	TOA			–	DT, ST	DL, SL
[26]					–	DT	ST, DL, SL
[14]					co-located sensor/source	ST	DT, DL, SL
[27], [28]	ambient sound	mic signal energy			co-located sensor/source	–	DL, SL
[16]	calibration signal	TOA		two-stage estimation	–	ST	DT, DL, SL
[17]–[19] [30]–[33]	ambient sound				–	DT	ST, DL, SL
proposed	ambient sound	TDOA			–	–	DT, DL, SL

TOA, the energy of the microphone signals can also be exploited for sensor and source localization, using a maximum likelihood optimization procedure [27]–[29]. The energy-based method does not need exact timing information, but usually assumes pairs of co-located sensors and sources, e.g., in laptops [27], [28]. In summary, most one-stage estimation algorithms assume that at least one set of timing information is known (either onset time or capture time) or impose additional geometrical constraints to simplify the optimization problem. Only [5] presents a general formulation.

A *two-stage* scheme that first estimates the timing information is more suitable for TOA measurements. After TOA correction, the scheme jointly estimates the sensor and source locations utilizing similar techniques to those employed in the one-stage scheme, e.g., [5], [13], [14]. An effective way of estimating timing information is to exploit the low-rank structure of the sensor and source locations, which all lie in the 3-dimensional space [30]. Specifically, a matrix consisting of the biased TOA measurements can be formulated which is supposed to be rank-3 if the bias is correctly compensated. With these constraints, the timing information can be estimated so that the mentioned matrix becomes rank-3 after compensation. Three algorithms have been proposed recently to solve this low-rank approximation problem, including alternating minimization [17], [19], nuclear truncation minimization [18], [31], [32], and structured total least squares [16]. The structured total least squares algorithm converges more than 100 times faster than the other two algorithms [16]. The two-stage estimation scheme is promising for the localization problem since by decomposing the problem into two stages, the optimization becomes easier.

In summary, the above-mentioned three low-rank approximation algorithms are suitable for TOA only, and assume that at least one timing information to be known (either onset time or capture time). For now, no progress has been reported on applying these algorithms to biased TDOA measurements. The proposed algorithm will fill this gap.

III. PROBLEM FORMULATION

We denote vectors and matrices in bold lowercase letters and bold uppercase letters, respectively. The operator $[\cdot]_i$ denotes the i-th element of a vector while $[\cdot]_{ij}$ denotes the (i,j)-th element of a matrix. The operator vec_c denotes vectorization of a matrix by concatenating its column vectors, while the operator vec_r denotes vectorization of a matrix by concatenating its row vectors.

A. Signal Model

Consider M independent acoustic sensors (microphones) and N sound sources (speakers) at unknown locations $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_M]_{3\times M}$ and $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N]_{3\times N}$, respectively. The emission onset times of the sources are unknown and represented as $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_N]^{\mathrm{T}}$, where the superscript $(\cdot)^{\mathrm{T}}$ denotes transpose. The unknown capture start times of the sensors are $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_M]^{\mathrm{T}}$. The TOA from the j-th source to the i-th sensor is defined as

$$t_{ij} = \frac{\|\boldsymbol{r}_i - \boldsymbol{s}_j\|}{c} + \eta_j - \delta_i, \tag{1}$$

where c is the speed of sound and $\|\cdot\|$ is the Euclidean norm. The TOAs for the combination of all the sensors and sources are represented as a matrix $\boldsymbol{T} = [t_{ij}]$ where $i = 1, \ldots, N$ and $j = 1, \ldots, M$. The TDOA from the j-th source to the first and the i-th sensor is defined as

$$\tau_{ij} = t_{ij} - t_{1j} = \frac{\|\boldsymbol{r}_i - \boldsymbol{s}_j\| - \|\boldsymbol{r}_1 - \boldsymbol{s}_j\|}{c} + \delta_1 - \delta_i, \tag{2}$$

where $\delta'_i = \delta_1 - \delta_i$ is the time offset between the first and the i-th sensor, and $d_{ij} = \|\boldsymbol{r}_i - \boldsymbol{s}_j\|$ is the distance between the i-th sensor and the j-th source (Fig. 1). The TDOAs for the combination of all the sensors and sources is represented as $\boldsymbol{T}_e = [\tau_{ij}]$ where $i = 1, \ldots, N$ and $j = 1, \ldots, M$. Based on (2), the first row of $\boldsymbol{T}_e$ consists of all zero elements, i.e., $\tau_{1,\cdot} = 0$. The time offsets of all the sensors with respect to the first sensor is $\boldsymbol{\delta}' = [\delta'_1, \ldots, \delta'_M]$, where $\delta'_1 = 0$.

We assume the sound sources are unknown (uncontrolled ambient sounds) and thus only TDOA information is available, which is biased by the unknown time offset. The goal of this paper is to estimate the sensor and source locations $\boldsymbol{R}$ and $\boldsymbol{S}$ from the biased TDOA measurements $\boldsymbol{T}_e$. Since the TDOA contains only the distance information, the estimation of the locations is not invariant against rotation, translation and reflection. Without loss of generality, we assume that the first sensor lies at the spatial origin $\boldsymbol{r}_1 = [0,0,0]^{\mathrm{T}}$ while the first source is confined to lie on the first positive axis, i.e., $\boldsymbol{s}_1 = [s_{x1}, 0, 0]^{\mathrm{T}}$ with $s_{x1} > 0$. We also assume that the onset time of the first source, η_1, is 0.

B. Existing Solutions

1) One-Stage Estimation: The sensor and source locations and the time offsets can be simultaneously estimated via maximum likelihood estimation. This is expressed as

$$\begin{aligned} &\hat{\boldsymbol{R}}, \hat{\boldsymbol{S}}, \hat{\boldsymbol{\delta}}' \\ &= \arg \min_{\boldsymbol{R},\boldsymbol{S},\boldsymbol{\delta}'} \sum_{i=2}^{M} \sum_{j=1}^{N} \left(\frac{\|\boldsymbol{r}_i - \boldsymbol{s}_j\| - \|\boldsymbol{r}_1 - \boldsymbol{s}_j\|}{c} + \delta_i' - \tau_{ij} \right)^2 . \end{aligned} \tag{3}$$

A gradient-based solution has been presented in [5]. However, with many unknown parameters, the minimization of (3) leads to a non-convex problem, which is sensitive to parameter initialization and can easily get stuck in local minima.

2) Two-Stage Estimation: The low-rank structure embedded in the TOA measurements can be exploited to estimate the timing and location information separately [17].

From (1) it can be derived that

$$\frac{\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i + \boldsymbol{s}_j^{\mathrm{T}}\boldsymbol{s}_j - 2\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{s}_j}{c^2} = t_{ij}^2 + \eta_j^2 + \delta_i^2 - 2(t_{ij}\eta_j - t_{ij}\delta_i + \eta_j\delta_i), \tag{4}$$

for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. Using the assumption $\eta_1 = 0$ and sequentially subtracting the corresponding equation for $i = 1$ and the equation for $j = 1$ from the general form (4), it follows

$$\begin{aligned} &\frac{-2(\boldsymbol{r}_i - \boldsymbol{r}_1)^{\mathrm{T}}(\boldsymbol{s}_j - \boldsymbol{s}_1)}{c^2} \\ &= t_{ij}^2 - t_{i1}^2 - t_{1j}^2 + t_{11}^2 + 2\delta_i(t_{ij} - t_{i1}) - 2\delta_1(t_{1j} - t_{11}) \\ &\quad - 2\eta_j(t_{ij} - t_{1j}) + 2\eta_j(\delta_1 - \delta_i) \end{aligned} \tag{5}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$. Equation (5) can be expressed in a matrix form as

$$\frac{-2\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}}}{c^2} = \boldsymbol{D} + \boldsymbol{U}. \tag{6}$$

The elements of $\bar{\boldsymbol{R}}_{3\times(M-1)}$, $\bar{\boldsymbol{S}}_{3\times(N-1)}$, $\boldsymbol{D}_{(M-1)\times(N-1)}$, and $\boldsymbol{U}_{(M-1)\times(N-1)}$ can be expressed as

$$\begin{aligned} \bar{\boldsymbol{R}}_{\cdot i-1} &= \boldsymbol{r}_i - \boldsymbol{r}_1, \\ \bar{\boldsymbol{S}}_{\cdot j-1} &= \boldsymbol{s}_j - \boldsymbol{s}_1, \\ \boldsymbol{D}_{i-1,j-1} &= t_{ij}^2 - t_{i1}^2 - t_{1j}^2 + t_{11}^2, \\ \boldsymbol{U}_{i-1,j-1} &= 2\delta_i(t_{ij} - t_{i1}) - 2\delta_1(t_{1j} - t_{11}) \\ &\quad - 2\eta_j(t_{ij} - t_{1j}) + 2\eta_j(\delta_1 - \delta_i), \end{aligned} \tag{7}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$.

Based on the above definitions, the matrices $\bar{\boldsymbol{R}}$ and $\bar{\boldsymbol{S}}$ are related to the sensor and source locations and their matrix ranks are both equal to 3. Thus, the rank of the left side of (6), $\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}}$, is equal to 3 and the same is true for the right side, $\boldsymbol{D} + \boldsymbol{U}$. This low-rank information can be utilized to estimate both the timing and geometrical information.

The matrix $\boldsymbol{D}$ is constructed from the TOA parameters t_{ij}, while the matrix $\boldsymbol{U}$ is constructed from the unknown parameters $\boldsymbol{\eta}$ and $\boldsymbol{\delta}$. Timing information estimation is essentially a low-rank matrix completion problem [36], i.e., the matrix $\boldsymbol{U}$ can be seen as a modification of $\boldsymbol{D}$ so that $\hat{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{U}$ is of rank 3 [17]. Low-rank approximation algorithms, such as alternative minimization [17], nuclear truncation minimization [18] and structured total least square (STLS) [16], have been proposed to solve this problem. After estimating the timing information and correcting the matrix, $\hat{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{U}$, the rank-3 of $\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}}$ is further used to estimate $\boldsymbol{R}$ and $\boldsymbol{S}$.

However, this solution is not directly applicable to our problem, where only TDOA is available and both onset times and capture times are unknown. The TDOA does not contain the desired low-rank structure as in (6). Existing low-rank approximation algorithms [16]–[18] assume either the onset times or the capture times to be known. To address these challenges, we propose a sensor and source localization method, which is described in Sections IV–VI.

IV. Pseudo TOA Matrix Construction

To employ the low-rank approximation method, a TOA matrix is needed which contains a low-rank structure as in (6). Since we only have the TDOA matrix $\boldsymbol{T}_e$ available, one possible solution is to convert $\boldsymbol{T}_e$ to a matrix containing the desired low-rank structure. To this end, we define a pseudo TOA matrix $\dot{\boldsymbol{T}}$ in the same form as $\boldsymbol{T}$ in (1), i.e.,

$$\dot{t}_{ij} = \frac{\|\boldsymbol{r}_i - \boldsymbol{s}_j\|}{c} + \dot{\eta}_j - \dot{\delta}_i, \tag{8}$$

where $\dot{\eta}_j$ and $\dot{\delta}_i$ are two unknown parameters termed as *pseudo onset time* and *pseudo capture time*, respectively. If we can find appropriate $\dot{\eta}_j$ and $\dot{\delta}_i$ that enable $\dot{\boldsymbol{T}} = \boldsymbol{T}_e$, the TDOA matrix would be interpreted as a TOA matrix.

To make $\dot{\boldsymbol{T}} = \boldsymbol{T}_e$, it requires $\dot{t}_{ij} = \tau_{ij}, \forall i, j$ or, more specifically,

$$\begin{bmatrix} \dot{t}_{11} & \cdots & \dot{t}_{1N} \\ \dot{t}_{21} & \cdots & \dot{t}_{2N} \\ \vdots & \ddots & \vdots \\ \dot{t}_{M1} & \cdots & \dot{t}_{MN} \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \\ \tau_{21} & \cdots & \tau_{2N} \\ \vdots & \ddots & \vdots \\ \tau_{M1} & \cdots & \tau_{MN} \end{bmatrix}. \tag{9}$$

Using (2) and (8) on $\dot{t}_{ij} = \tau_{ij}$, it follows that

$$\dot{\eta}_j - \dot{\delta}_i = -\frac{\|\boldsymbol{r}_1 - \boldsymbol{s}_j\|}{c} + \delta_i', \tag{10}$$

for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. It can be easily verified that the two sets of pseudo parameters $\dot{\eta}_j$ ($j = 1, \ldots, N$) and $\dot{\delta}_i$ ($i = 1, \ldots, M$) can be uniquely determined by solving a group of (10) if the sensor locations $\boldsymbol{R}$, the source locations $\boldsymbol{S}$ and the time offsets $\boldsymbol{\delta}'$ are known. This demonstrates theoretically that there do exist appropriate $\dot{\eta}_j$ and $\dot{\delta}_i$ that enable $\dot{\boldsymbol{T}} = \boldsymbol{T}_e$. In practice, these two parameters are unknown and will have to be estimated from $\boldsymbol{T}_e$.

Thus, by introducing two pseudo parameters, the TDOA matrix $\boldsymbol{T}_e$ can be interpreted as the pseudo TOA matrix $\dot{\boldsymbol{T}}$. Obviously, $\dot{\boldsymbol{T}}$ and its pseudo parameters $\dot{\eta}_j$ and $\dot{\delta}_i$ are different from the original TOA, onset time and capture time, which are defined in (1). However, with similar structure as the original TOA matrix, $\dot{\boldsymbol{T}}$ can be exploited for low-rank approximation. The next task will be to estimate $\dot{\boldsymbol{\delta}} = [\dot{\delta}_1, \ldots, \dot{\delta}_M]^{\mathrm{T}}$ and $\dot{\boldsymbol{\eta}} = [\dot{\eta}_2, \ldots, \dot{\eta}_N]^{\mathrm{T}}$ and to recover $\boldsymbol{R}$ and $\boldsymbol{S}$.

Considering that the concepts of original TOA, onset time and capture time will not be used hereafter, we remove the accent $(\dot{\ })$ of $\dot{\boldsymbol{T}}$, $\dot{\boldsymbol{\delta}}$ and $\dot{\boldsymbol{\eta}}$, and also call them TOA, onset time and capture time, without introducing ambiguities.

V. Timing Information Estimation

In this section, we propose a Gauss-Newton low-rank approximation algorithm to estimate the unknown parameters $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ from the pseudo TOA matrix $\boldsymbol{T}$.

As shown in Section III-B2, the timing information estimation can be seen as a low-rank matrix completion problem (6). We solve this problem under a structured total least-square framework [16]. The matrices $\boldsymbol{D}, \boldsymbol{U}$, and $\hat{\boldsymbol{D}}$ are each partitioned into two blocks, such that the first submatrix contains the first three columns while the second submatrix contains the remaining columns. This is expressed as

$$\boldsymbol{D} = [\boldsymbol{A}_{M-1\times 3} \,|\, \boldsymbol{B}_{M-1\times N-4}], \tag{11}$$

$$\boldsymbol{U} = [\boldsymbol{F}_{M-1\times 3} \,|\, \boldsymbol{G}_{M-1\times N-4}], \tag{12}$$

$$\hat{\boldsymbol{D}} = [(\boldsymbol{A}+\boldsymbol{F}) \,|\, (\boldsymbol{B}+\boldsymbol{G})], \tag{13}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{F}$ and $\boldsymbol{G}$ are the partitioned submatrices. The objective is to find an appropriate $\boldsymbol{U}$ (consisting of $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$) and an additional matrix $\boldsymbol{X}$ so that

$$(\boldsymbol{A}+\boldsymbol{F})\boldsymbol{X} = \boldsymbol{B}+\boldsymbol{G}. \tag{14}$$

If (14) holds, the matrix $\boldsymbol{B}+\boldsymbol{G}$ will lie in the range of the matrix $\boldsymbol{A}+\boldsymbol{F}$. It follows that

$$\operatorname{rank}([(\boldsymbol{A}+\boldsymbol{F}) \,|\, (\boldsymbol{B}+\boldsymbol{G})]) = \operatorname{rank}(\boldsymbol{A}+\boldsymbol{F}) = 3. \tag{15}$$

Similarly to [16], the STLS cost function is defined as

$$\boxed{\xi(\boldsymbol{\delta}, \boldsymbol{\eta}, \boldsymbol{X}) = \|\boldsymbol{U}\|_F^2 + \lambda^2 \|(\boldsymbol{A}+\boldsymbol{F})\boldsymbol{X} - \boldsymbol{B} - \boldsymbol{G}\|_F^2} \tag{16}$$

where $\|\cdot\|_F$ denotes the F-norm. The first term on the right side of (16) is introduced to prevent algorithm divergence; the second term corresponds to the condition (14) and is emphasized with a large penalty λ. In [16] a simplified solution is given, which however assumes that the onset time $\boldsymbol{\eta}$ is known. This solution is not feasible for our case, where both $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ are unknown. We present a complete and general solution below.

Let us define the unknown parameters $\boldsymbol{\delta}$, $\boldsymbol{\eta}$, and $\boldsymbol{X}$ as a column vector $\boldsymbol{\rho} = [\boldsymbol{\delta}^{\mathrm{T}} \quad \boldsymbol{\eta}^{\mathrm{T}} \quad \boldsymbol{x}^{\mathrm{T}}]^{\mathrm{T}}$, where $\boldsymbol{x} = \operatorname{vec}_c(\boldsymbol{X}) = [[\boldsymbol{X}]_{11}, [\boldsymbol{X}]_{21}, \ldots, [\boldsymbol{X}]_{12}, [\boldsymbol{X}]_{22}, \ldots]^{\mathrm{T}}$. The cost function (16) can be further written as

$$\begin{aligned}\xi(\boldsymbol{\rho}) &= \sum_{i=1}^{M-1}\sum_{j=1}^{N-1}[\boldsymbol{U}]_{ij}^2 + \lambda^2 \sum_{i=1}^{M-1}\sum_{j=1}^{N-4}[(\boldsymbol{A}+\boldsymbol{F})\boldsymbol{X} - \boldsymbol{B} - \boldsymbol{G}]_{ij}^2 \\ &= \sum_{q=1}^{Q}\zeta_q^2,\end{aligned} \tag{17}$$

where $Q = (M-1)(N-1) + (M-1)(N-4)$ and $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_A^{\mathrm{T}} \quad \lambda\boldsymbol{\zeta}_B^{\mathrm{T}}]^{\mathrm{T}} = [\zeta_1, \ldots, \zeta_Q]^{\mathrm{T}}$, with the two column vectors

$$\begin{cases}\boldsymbol{\zeta}_A = \operatorname{vec}_c(\boldsymbol{U}) \\ \boldsymbol{\zeta}_B = \operatorname{vec}_c((\boldsymbol{A}+\boldsymbol{F})\boldsymbol{X} - \boldsymbol{B} - \boldsymbol{G})\end{cases}. \tag{18}$$

Based on (17), the minimization of ξ is a nonlinear least square problem, which can be solved with the well-known Gauss-Newton algorithm [35]. Suppose the Jacobian matrix

$$\boldsymbol{J} = \frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\rho}} \tag{19}$$

can be obtained. The unknown $\boldsymbol{\rho}$ can be estimated in an iterative way as

$$\boldsymbol{\rho}^{(m+1)} = \boldsymbol{\rho}^{(m)} - ({\boldsymbol{J}^{(m)}}^{\mathrm{T}}\boldsymbol{J}^{(m)})^{-1}{\boldsymbol{J}^{(m)}}^{\mathrm{T}}\boldsymbol{\zeta}^{(m)}, \tag{20}$$

where $\boldsymbol{\rho}^{(m)}$, $\boldsymbol{J}^{(m)}$ and $\boldsymbol{\zeta}^{(m)}$ denote the parameters obtained in the m-th iteration. The computation of the Jacobian matrix is presented in Appendix A.

To tackle the local minima problem, we reduce the initialization space by computing the boundary of the parameters (Section V-A). We also use multiple initializations to increase the probability of reaching the global minimum (Section V-B).

A. Parameter Boundary

We derive the boundary of the unknown parameters by using the definitions of $\boldsymbol{T}$ and t_{ij} in (8)–(10), and the assumptions $\boldsymbol{r}_1 = [0 \quad 0 \quad 0]^{\mathrm{T}}$ and $\eta_1 = 0$.

Using $t_{11} = 0$, it follows that

$$\frac{d_{11}}{c} + \delta_1 = 0 \Rightarrow \delta_1 = -\frac{\|\boldsymbol{s}_1\|}{c}. \tag{21}$$

Using $t_{1j} = 0$ for $j = 2, \ldots, N$, it follows that

$$\frac{d_{11}}{c} + \delta_1 + \eta_j = 0 \Rightarrow \eta_j = \frac{\|\boldsymbol{s}_1\| - \|\boldsymbol{s}_j\|}{c} \tag{22}$$

$$\Rightarrow -\frac{\|\boldsymbol{s}_j\|}{c} < \eta_j < \frac{\|\boldsymbol{s}_1\|}{c}. \tag{23}$$

Averaging on the i-th ($i = 2, \ldots, M$) row of $\boldsymbol{T}$, it follows

$$\frac{1}{N}\sum_{j=1}^{N} t_{ij} = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{d_{ij}}{c} + \delta_i + \eta_j\right) \tag{24}$$

$$\Rightarrow \delta_i = \frac{1}{N}\sum_{j=1}^{N} t_{ij} - \frac{1}{N}\sum_{j=1}^{N}\left(\frac{d_{ij}}{c} + \eta_j\right). \tag{25}$$

Using (22), it follows that

$$\sum_{j=1}^{N}\left(\frac{d_{ij}}{c}+\eta_j\right)=\frac{d_{i1}}{c}+\sum_{j=2}^{N}\left(\frac{d_{ij}+\|\boldsymbol{s}_1\|-\|\boldsymbol{s}_j\|}{c}\right). \quad (26)$$

Using (26) in (25), it follows that

$$\frac{1}{N}\left(\sum_{j=1}^{N}t_{ij}-\frac{d_{i1}}{c}-\sum_{j=2}^{N}\left(\frac{d_{ij}+\|\boldsymbol{s}_1\|}{c}\right)\right) < \delta_i < \frac{1}{N}\left(\sum_{j=1}^{N}t_{ij}+\sum_{j=2}^{N}\left(\frac{\|\boldsymbol{s}_j\|}{c}\right)\right). \quad (27)$$

Suppose the maximum distance between a sensor and a source is known to be $d_{\max}$. Using $d_{\max}$ in (21), (23) and (27), the boundary of the parameters can be obtained as

$$\boxed{\begin{cases} -\frac{d_{\max}}{c} < \eta_j < \frac{d_{\max}}{c} \\ \frac{1}{N}\sum\limits_{j=1}^{N} t_{ij} - \frac{d_{\max}}{c} < \delta_1 < \frac{1}{N}\sum\limits_{j=1}^{N} t_{ij} + \frac{d_{\max}}{c} \\ \frac{1}{N}\sum\limits_{j=1}^{N} t_{ij} - \frac{(2N-1)d_{\max}}{Nc} < \delta_i < \frac{1}{N}\sum\limits_{j=1}^{N} t_{ij} + \frac{(N-1)d_{\max}}{cN} \end{cases}} \quad (28)$$

This is an interesting result since, based on (28), the values of the onset times and capture times are only related to the locations of sensors and sources (more specifically $d_{\max}$) and a constant term $\frac{1}{N}\sum_{j=1}^{N} t_{ij}$. With (28), the initialization space can be significantly reduced compared to (unconstrained) random initialization. For example, assuming $d_{\max} = 10$ m, $c = 342$ m/s, and $N = 10$, the parameter space can be initialized as $\delta_1 - \frac{1}{N}\sum_{j=1}^{N} t_{ij} \in [-0.03, 0.03]$ s, $\delta_i - \frac{1}{N}\sum_{j=1}^{N} t_{ij} \in [-0.06, 0.03]$ s, and $\eta_j \in [-0.03, 0.03]$ s.

B. Multiple Initializations

To overcome the local minima problem we use multiple initializations, a scheme that is suitable for the low-rank approximation algorithm. First, the Gauss-Newton algorithm can converge at an approximately quadratic rate, which takes only a few iterations to reach global or local minima [35]. This makes it computationally feasible to try multiple initializations. Second, we experimentally observe that the difference between the global and local minima is very evident, making it easy to detect whether a global minimum has been reached.

Fig. 2 gives an example of the convergence behaviour of the Gauss-Newton low-rank approximation algorithm, using 10 sources and 10 sensors randomly distributed with $d_{\max} = 10$ m. The start times and capture times are chosen randomly and uniformly from the range $[-1, 1]$ s. The algorithm uses 1000 random initializations uniformly chosen from the range defined by (28). The results are shown in terms of cost function $\|\boldsymbol{\xi}_B\|_F^2$, where $\boldsymbol{\xi}_B$ is defined in (18), and estimation error ($\|\boldsymbol{\rho} - \tilde{\boldsymbol{\rho}}\|$ for the true value $\boldsymbol{\rho}$ and the estimated value $\tilde{\boldsymbol{\rho}}$). Fig. 2(a) depicts the final cost function obtained by each initialization. Three types of results can be achieved: global convergence, local convergence, and divergence. Their difference can be clearly observed: the cost function values below 10^{-20} in case of global convergence, around 10^{-10} in case of local convergence, and over 10^{10} in case of divergence. This makes it easy to detect the convergence status of the algorithm. Accordingly, Fig. 2(b) depicts the final estimation error by each initialization. The three types of convergence result in different estimation errors: these errors are around 10^{-15} s in case of global convergence, around 0.1 s in case of local convergence and over 10^{10} s in case of divergence. Fig. 2(c) and (d) depict typical cost function and estimation error curves during the Gauss-Newton iteration. With quadric convergence rate, the algorithm can converge either locally or globally within 100 iterations.

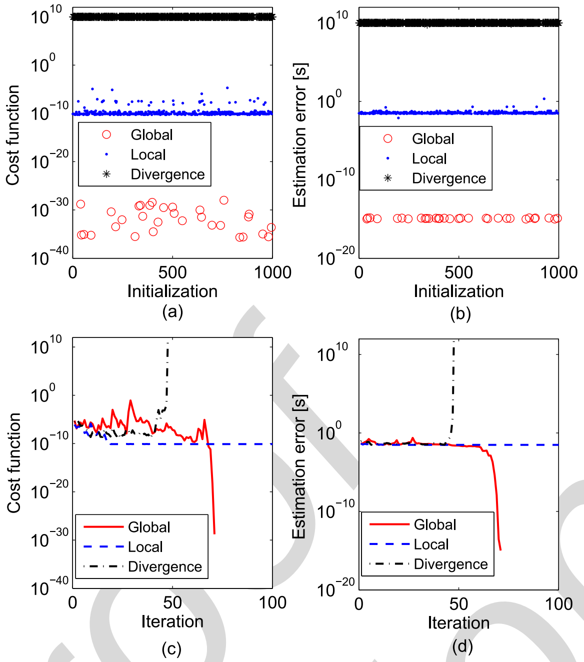


Fig. 2. Convergence results in terms of (a) cost function and (b) estimation error by the Gauss-Newton low-rank approximation algorithm with 1000 initializations ($M = N = 10$). Typical iteration curves of (c) cost function and (d) estimation error for global converge, local convergence, and divergence.

The description of the timing information estimation algorithm is given in Algorithm 1. The input $\boldsymbol{T}$ equals the TDOA $\boldsymbol{T}_e$. We try multiple initializations until the algorithm reaches global convergence. For each initialization, $\boldsymbol{\delta}^{(0)}$ and $\boldsymbol{\eta}^{(0)}$ are randomly chosen within the range (28).

The value of the cost function threshold (ζ_{T2} in Algorithm 1) is chosen based on the convergence results obtained with clean TDOA measurements. This threshold can be easily determined since the final cost functions for local and global convergence show evident differences (e.g., we choose a threshold of 10^{-15} according to Fig. 2(c)). For noisy inputs, the final cost function for global convergence increases with TDOA errors, making the difference between local and global convergence less evident. Thus, the optimal threshold may vary with the amount of noise. However, we will use the same threshold for all scenarios. There are two reasons for that. First, since the final cost function for global convergence increases with TDOA errors, the algorithm will reach global convergence once the cost function is below the threshold, which is determined for clean TDOA measurements. Second, if in case of noisy inputs the final cost function for global convergence is higher than the threshold, the algorithm will run until reaching the maximum number of initializations and choose the solution with the minimum cost function. This multi-initialization scheme can reduce the influence of a too-low threshold.

Algorithm 1 Gauss-Newton low-rank approximation for timing information estimation

```
Input: $\boldsymbol{T}$
Output: $\boldsymbol{\rho}_o = \{\boldsymbol{\delta}, \boldsymbol{\eta}, \boldsymbol{X}\}$
Parameters: $\zeta_{T1}$: threshold for divergence
            $\zeta_{T2}$: threshold for global convergence
            $d_{\rho T}$: stopping threshold for Gauss-Newton iterations
            $m_{T1}$: maximum number of initializations
            $m_{T2}$: maximum number of iterations
            $\lambda$: penalty constant in (16)
            $d_{\max}$: maximum sensor-source distance in (28)
$f_1 = 0$; $m_1 = 0$; $\zeta_o = \mathrm{Inf}$;
while $f_1 == 0$ do
    Initialize $\boldsymbol{\delta}^{(0)}$, $\boldsymbol{\eta}^{(0)}$; (28)
    Compute $\boldsymbol{D}$, $\boldsymbol{A}$, $\boldsymbol{B}$; (7)(11)
    $\boldsymbol{X}^{(0)} = (\boldsymbol{A}+\boldsymbol{F})^{-1}(\boldsymbol{B}+\boldsymbol{G})$; $f_2 = 0$; $m_2 = 0$;
    while $f_2 == 0$ do
        Compute $\boldsymbol{U}^{(m_2)}$, $\boldsymbol{F}^{(m_2)}$, $\boldsymbol{G}^{(m_2)}$; (7)
        Compute $\boldsymbol{\zeta}^{(m_2)} = \begin{bmatrix} \boldsymbol{\zeta}_A^{(m_2)} \\ \lambda\boldsymbol{\zeta}_B^{(m_2)} \end{bmatrix}^{\mathrm{T}}$; (17)
        Compute $\boldsymbol{J}^{(m_2)}$; (19)
        Update $\boldsymbol{\rho}^{(m_2+1)}$; (20)
        if $\|\boldsymbol{\zeta}_B^{(m_2)}\|_F^2 > \zeta_{T1}$ OR $\|\boldsymbol{\Delta\rho}\| < d_{\rho T}$ OR
        $m_2 == m_{T2}$
            then $f_2 = 1$;
        end if
        $m_2 \leftarrow m_2 + 1$;
    end while
    if $\|\boldsymbol{\zeta}_B^{(m_2-1)}\|_F^2 < \zeta_o$ then
        $\zeta_o = \|\boldsymbol{\zeta}_B^{(m_2-1)}\|_F^2$; $\boldsymbol{\rho}_o = \boldsymbol{\rho}^{(m_2-1)}$;
    end if
    if $\zeta_o < \zeta_{T2}$ OR $m_1 == m_{T1}$; then $f_1 = 1$; end if
    $m_1 \leftarrow m_1 + 1$;
end while
```

VI. Geometrical Information Estimation

Building on the work in [13], [17], in this section we present a Gauss-Newton joint localization algorithm to estimate the sensor and source locations, using the timing information obtained in Section V.

Given the estimated $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$, (6) becomes $\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}} = -\frac{c^2}{2}\hat{\boldsymbol{D}}$. Since $\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}}$ is of rank 3, it can be decomposed via singular value decomposition (SVD) as

$$\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}} = \boldsymbol{D}_L\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}}, \tag{29}$$

where $\boldsymbol{\Sigma}_3$ keeps only the three largest singular values while $\boldsymbol{D}_L$ and $\boldsymbol{D}_R$ consist of the corresponding left-singular and right-singular vectors, respectively. With (29), $\bar{\boldsymbol{R}}$ and $\bar{\boldsymbol{S}}$ can be calculated as

$$\begin{cases} \bar{\boldsymbol{R}}^{T} = \boldsymbol{D}_L\boldsymbol{C} \\ \bar{\boldsymbol{S}} = \boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}} \end{cases}, \tag{30}$$

where $\boldsymbol{C}$ is an arbitrary 3×3 matrix which is invertible. Using the assumptions $\boldsymbol{r}_1 = [0,0,0]^{\mathrm{T}}$, $\boldsymbol{s}_1 = [s_{x1},0,0]^{\mathrm{T}}$, the sensor and source locations can be calculated as

$$\begin{cases} \boldsymbol{r}_i = \bar{\boldsymbol{r}}_i \\ \boldsymbol{s}_j = \bar{\boldsymbol{s}}_j + [\, s_{x1} \quad 0 \quad 0 \,]^{\mathrm{T}} \end{cases}. \tag{31}$$

For localization, we only need to estimate 10 parameters regardless of the number of sensors and sources. The 10 unknown parameters are represented as $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{c} \\ s_{x1} \end{bmatrix}$ where $\boldsymbol{c} = \mathrm{vec}_c(\boldsymbol{C}) = [[\boldsymbol{C}]_{11}, [\boldsymbol{C}]_{21}, \ldots, [\boldsymbol{C}]_{33}]^{\mathrm{T}}$.

To estimate $\boldsymbol{\beta}$, we rewrite the relationship (4) as

$$\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i + \boldsymbol{s}_j^{\mathrm{T}}\boldsymbol{s}_j - 2\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{s}_j = g_{ij}, \tag{32}$$

with $g_{ij} = c^2(t_{ij}^2 + \eta_j^2 + \delta_i^2 - 2(t_{ij}\eta_j - t_{ij}\delta_i + \eta_j\delta_i))$. We further have

$$\begin{cases} \boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i - \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{r}_1 - 2(\boldsymbol{r}_i^{\mathrm{T}} - \boldsymbol{r}_1^{\mathrm{T}})\boldsymbol{s}_j = g_{ij} - g_{1j} \\ \boldsymbol{s}_j^{\mathrm{T}}\boldsymbol{s}_j - \boldsymbol{s}_1^{\mathrm{T}}\boldsymbol{s}_1 - 2\boldsymbol{r}_i^{\mathrm{T}}(\boldsymbol{s}_j - \boldsymbol{s}_1) = g_{ij} - g_{i1} \end{cases}. \tag{33}$$

Using (31) in (33), it follows that

$$\begin{cases} \bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{r}}_i - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}(\bar{\boldsymbol{s}}_j + \boldsymbol{s}_1) = g_{ij} - g_{1j} \\ \bar{\boldsymbol{s}}_j^{\mathrm{T}}\bar{\boldsymbol{s}}_j + 2\bar{\boldsymbol{s}}_j^{\mathrm{T}}\boldsymbol{s}_1 - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{s}}_j = g_{ij} - g_{i1} \end{cases}. \tag{34}$$

Using (6) in (34), it follows that

$$\boxed{\begin{cases} \bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{r}}_i - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}\boldsymbol{s}_1 = \bar{g}_i \\ \bar{\boldsymbol{s}}_j^{\mathrm{T}}\bar{\boldsymbol{s}}_j + 2\bar{\boldsymbol{s}}_j^{\mathrm{T}}\boldsymbol{s}_1 = \bar{\bar{g}}_j \end{cases}} \tag{35}$$

where

$$\begin{cases} \bar{g}_{ij} = g_{ij} - g_{1j} - c^2[\hat{\boldsymbol{D}}]_{ij} \\ \bar{\bar{g}}_{ij} = g_{ij} - g_{i1} - c^2[\hat{\boldsymbol{D}}]_{ij} \end{cases}, \tag{36}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$. It can be easily verified that $\bar{g}_{ij}$ has the same value for any j while $\bar{\bar{g}}_{ij}$ has the same value for any i. Thus they can be represented as $\bar{g}_i$ and $\bar{\bar{g}}_j$.

The $M+N-2$ equations in (35) can be used to estimate $\boldsymbol{\beta}$. The cost function is defined as

$$\begin{aligned}\varsigma(\boldsymbol{\beta}) &= \sum_{i=2}^{M}(\bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{r}}_i - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}\boldsymbol{s}_1 - \bar{g}_i)^2 + \sum_{j=2}^{N}(\bar{\boldsymbol{s}}_j^{\mathrm{T}}\bar{\boldsymbol{s}}_j + 2\bar{\boldsymbol{s}}_j^{\mathrm{T}}\boldsymbol{s}_1 - \bar{\bar{g}}_j)^2\\ &= \sum_{p=1}^{M+N-2}\nu_p^2, \end{aligned} \tag{37}$$

where $\boldsymbol{\nu} = [\boldsymbol{\nu}_A^{\mathrm{T}} \quad \boldsymbol{\nu}_B^{\mathrm{T}}]^{\mathrm{T}} = [\nu_1, \ldots, \nu_{M+N-2}]^{\mathrm{T}}$, with

$$\begin{cases}[\boldsymbol{\nu}_A]_{i-1} = \bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{r}}_i - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}\boldsymbol{s}_1 - \bar{g}_i\\ [\boldsymbol{\nu}_B]_{j-1} = \bar{\boldsymbol{s}}_j^{\mathrm{T}}\bar{\boldsymbol{s}}_j + 2\bar{\boldsymbol{s}}_j^{\mathrm{T}}\boldsymbol{s}_1 - \bar{\bar{g}}_j\end{cases}, \tag{38}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$.

The minimization of ς is a nonlinear least square problem which can be solved with the Gauss-Newton algorithm. Suppose the Jacobian matrix

$$\boldsymbol{I} = \frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\beta}} \tag{39}$$

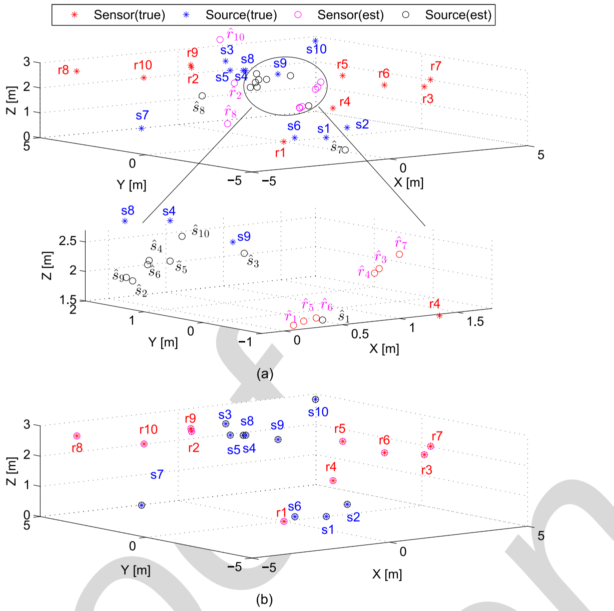
can be obtained. $\boldsymbol{\beta}$ can be estimated iteratively as

$$\boldsymbol{\beta}^{(m+1)} = \boldsymbol{\beta}^{(m)} - (\boldsymbol{I}^{(m)^{\mathrm{T}}}\boldsymbol{I}^{(m)})^{-1}\boldsymbol{I}^{(m)^{\mathrm{T}}}\boldsymbol{\nu}^{(m)} \tag{40}$$

where $\boldsymbol{I}^{(m)}$, $\boldsymbol{\beta}^{(m)}$ and $\boldsymbol{\nu}^{(m)}$ denote the parameters obtained in the m-th iteration. The detailed derivation of the Jacobian matrix is given in Appendix B.

Compared with [13], [17], which only consider $\boldsymbol{\nu}_A$, the proposed algorithm considers both $\boldsymbol{\nu}_A$ and $\boldsymbol{\nu}_B$, i.e., more observations are used. This may lead to better convergence behaviour. Similarly to Algorithm 1, we use a multi-initialization scheme. The description of the Gauss-Newton joint localization algorithm is given in Algorithm 2. The elements in $\boldsymbol{\beta}^{(0)}$ are initialized as normally distributed random numbers.

The convergence behaviour of the Gauss-Newton joint localization algorithm is closely related to the accuracy of timing information estimation. If the low-rank approximation algorithm only converges locally, (29) does not exactly hold and the joint localization algorithm converges locally or even diverges. If the low-rank approximation algorithm converges globally with accurate timing information estimation, the joint localization algorithm can globally converge easily with a multi-initialization scheme. Fig. 3 gives an example of sensor and source localization using the same data as Fig. 2. In Fig. 3(a), the low-rank approximation algorithm converges locally and so does the joint localization algorithm. In Fig. 3(b), the low-rank approximation algorithm converges globally and the joint localization algorithm can recover the sensor and source locations accurately.



(a)

(b)

Fig. 3. Sensor and source localization by the proposed two-stage method ($M = N = 10$). The Gauss-Newton low-rank approximation algorithm and the Gauss-Newton joint localization algorithm both converge (a) locally and (b) globally.

Algorithm 2 Gauss-Newton joint localization algorithm for geometrical information localization

```
Input: $\boldsymbol{T}$, $\boldsymbol{\delta}$, $\boldsymbol{\eta}$
Output: $\boldsymbol{R}$, $\boldsymbol{S}$, $\boldsymbol{\beta}_o = \{\boldsymbol{C}, s_{x1}\}$
Parameters: $\nu_{T1}$: threshold for divergence
        $\nu_{T2}$: threshold for global convergence
        $d_{\beta T}$: stopping threshold for Gauss-Newton iterations
        $m_{T3}$: maximum number of initializations
        $m_{T4}$: maximum number of iterations
Compute $\hat{\boldsymbol{D}}$; (13)
Compute $\boldsymbol{D}_L$, $\boldsymbol{D}_R$, $\boldsymbol{\Sigma}_3$ via SVD; (29)
$f_3 = 0$; $m_3 = 0$; $\nu_o = \mathrm{Inf}$;
while $f_3 == 0$ do
    Initialize $\boldsymbol{\beta}^{(0)}$; $f_4 = 0$; $m_4 = 0$;
    while $f_4 == 0$ do
        Compute $\bar{\boldsymbol{R}}^{(m_4)}$, $\bar{\boldsymbol{S}}^{(m_4)}$; (30)
        Compute $\boldsymbol{\nu}^{(m_4)}$, $\boldsymbol{I}^{(m_4)}$; (38)(39)
        Update $\boldsymbol{\beta}^{(m_4+1)}$; (40)
        if $\|\boldsymbol{\nu}^{(m_4)}\|_F^2 > \nu_{T1}$ OR $\|\boldsymbol{\Delta\beta}\| < d_{\beta T}$ OR $m_4 == m_{T4}$
            then $f_4 = 1$;
        end if
        $m_4 \leftarrow m_4 + 1$;
    end while
    if $\|\boldsymbol{\nu}^{(m_4-1)}\|_F^2 < \nu_o$ then
        $\nu_o = \|\boldsymbol{\nu}(m_4 - 1)\|_F^2$; $\boldsymbol{\beta}_o = \boldsymbol{\beta}^{(m_4-1)}$;
    end if
    if $\nu_o < \nu_{T2}$ OR $m_3 == m_{T3}$; then $f_3 = 1$; end if
    $m_3 \leftarrow m_3 + 1$;
end while
Compute $\boldsymbol{R}$, $\boldsymbol{S}$; (31)
```

VII. Experimental Results

A. Experiment Setup

The proposed algorithm consists of two stages: the Gauss-Newton low-rank approximation algorithm for timing information estimation and the Gauss-Newton joint localization algorithm for geometrical information estimation. The first sub-algorithm is sensitive to local minima. Five experiments are thus carried out: the first experiment examines the convergence property of the timing information estimation algorithm; the second experiment compares the performance of the proposed algorithm with existing algorithms; the third investigates the robustness of the proposed algorithm to noisy TDOA measurements; the fourth evaluates the performance with realistic TDOA data estimated from simulated ambient sounds; the last one presents an example with real-recorded data.

Monte Carlo simulations are used to generate testing data. We test all the possible combinations of different numbers of sensors (M) and sources (N), where M is chosen from [5:15, 20] and N is chosen from [5:15, 20, 25]. For each (M, N), we implement different realizations. In each realization, the locations of the sensors and sources are uniformly distributed inside an enclosure of 10 m × 10 m × 3 m with $d_{\max} = 10$ m, whereas the onset times and capture times are chosen randomly and uniformly from the range $[-1, 1]$ s.

A fair comparison of the proposed algorithm with existing algorithms is problematic since most existing approaches assume some prior knowledge of the timing or location information. The closest approach to ours is the one proposed by Ono [5] which does not consider any prior knowledge. Three algorithms are therefore considered: the one-stage algorithm (`Ono`) [5], the proposed two-stage algorithm (`Proposed`), and the `Proposed` algorithm refined by the `Ono` algorithm (`Proposed + refinement`). As discussed in Section VI, the `Proposed` algorithm may not recover the locations if the timing information is not estimated accurately. However, the estimated locations are usually closer to the true values than random initializations. Using them as an initial guess for the `Ono` algorithm, which estimates all the parameters simultaneously, may lead to a better estimation. The parameters used in the `Proposed` algorithm are listed in Table II. These parameters will be used throughout the experiment unless otherwise stated. In the `Ono` algorithm, the maximum iteration number is set to 2×10^5.

B. Convergence Analysis

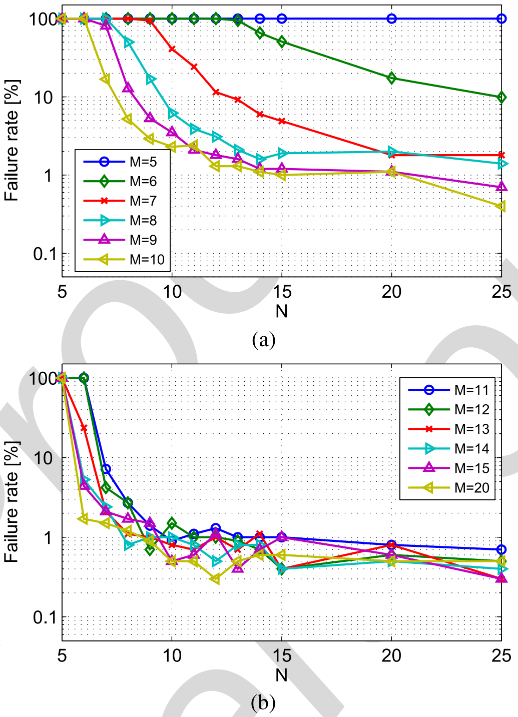
To investigate the convergence behaviour of the timing information estimation (Gauss-Newton low-rank approximation) algorithm, we test all the possible combinations of M and N. For each (M, N) configuration, we implement 1000 realizations. To reach global convergence, we try at most 10000 initializations in each realization. As described in Algorithm 1, a global convergence is detected when $\zeta_o < \zeta_{T2}$. If for one realization the algorithm can reach global convergence within 10000 initializations, we regard this realization as success and record the corresponding number of initializations as I_G. Otherwise, we regard it as failure. The failure rate is defined as

$$R_F(M, N) = \frac{N_F(M, N)}{1000} \times 100\%, \tag{41}$$

where $N_F(M, N)$ denotes the number of failed realizations for a specific (M, N) configuration.

TABLE II
Parameters Used in the Proposed Algorithm.

Parameter	Value	Parameter	Value
m_{T1}	1000	m_{T3}	50
m_{T2}	100	m_{T4}	100
ζ_{T1}	10^5	ν_{T1}	10^5
ζ_{T2}	10^{-15}	ν_{T1}	10^{-15}
$d_{\rho T}$	10^{-9}	$d_{\beta T}$	10^{-9}
λ	10^9	$d_{\max}$	10 m



(a)

(b)

Fig. 4. Failure rate for the timing information estimation when varying the number of sensors (M) and sources (N). (a) $M \in [5, 10]$. (b) $M \in [11, 20]$.

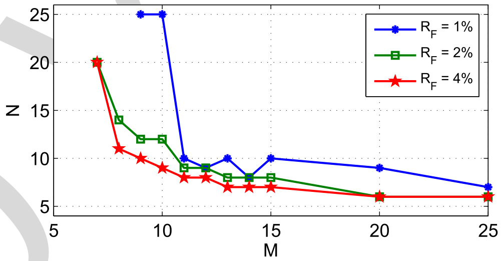


Fig. 5. Required minimum number of sensors (M) and sources (N) by the timing information estimation algorithm to achieve a failure rate of 1%, 2%, and 4%.

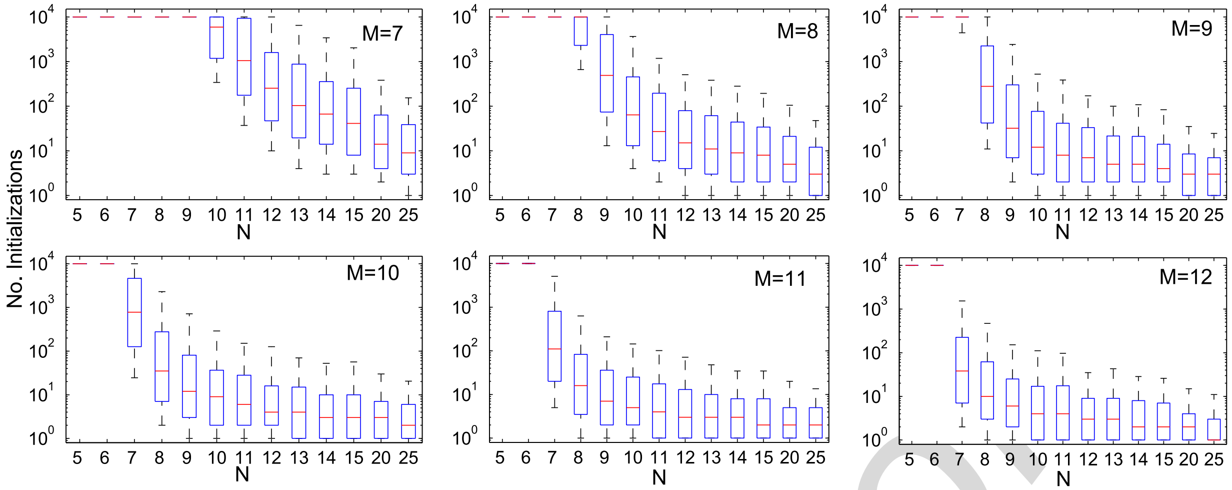


Fig. 6. Number of initializations (I_G) tried by the timing information estimation algorithm before reaching global convergence.

Each curve in Fig. 4 depicts the failure rate for a fixed M and variable N. Overall, the algorithm shows a high failure rate for small M and N, whereas the failure rate drops significantly when M and N increase. For the non-convex cost function (16) an over-constrained case with more observations than unknown parameters might reduce the number of local minima and lead to better convergence. The algorithm fails when $M = 5$, regardless of the value of N. When $M = 6$, the algorithm reaches global convergence even if the failure rate is still high. When $M \geq 7$, it becomes easier to reach global convergence. A fairly low failure rate (e.g., $< 2\%$) can be achieved if N is large enough. Moreover, the minimum N that is required to reach a same failure rate decreases when increasing M (cf. Fig. 5). When $M \geq 11$, the failure-rate curves become similar to each other. A low failure rate can be easily reached. In addition to this, when M and N are both large, the failure rate is slightly higher than 0, e.g., $R_F = 0.5\%$ for $M = 20$, $N = 25$.

Fig. 5 gives the minimum N that is required for reaching a defined failure rate ($R_F < 1\%, 2\%, 4\%$) for different M. It is observed that the lower the failure rate, the larger M and N are required. For the same failure rate the required minimum N decreases with M. This can be observed when $M > 10$.

Fig. 6 provides box-plots of the number of initializations (I_G) tried by the algorithm before reaching global convergence in 1000 realizations. The bottom and top of the box denote the 0.1 and 0.9 quantile, respectively. The line in the middle of the box denotes the median value. The bars represent the extreme values excluding the outliers, which are not shown in the figure.

In global, I_G decreases when M and N increase. This can be clearly observed from the variation of the median value of I_G in each panel, where M is fixed and N is varying. It can be further observed in each panel that, for a fixed M, the median value of I_G equals 10000 when N is small (the algorithm diverges) and then decreases significantly with increasing N. The decrease slows when N is large enough.

The span of the box (including 10% – 90% of the data) indicates that the performance of the algorithm may vary with different realizations of an (M, N) configuration. The height of the box is also related to M and N, and typically decreases when M and N increase. For instance, the box for $M = N = 8$ spans an interval [2000, 10000] while the box for $M = N = 12$ spans an interval [1, 10]. This indicates that it is easier for the algorithm to converge globally when M and N are large enough.

Fig. 6 can provide a reference for determining the maximum number of initializations (m_{T1}), a parameter used in Algorithm 1. For instance, we choose $m_{T1} = 1000$ in the remaining experiment since in most (M, N) configurations the algorithm can converge globally within 1000 initializations.

C. Performance Comparison

The testing data is generated using an equal number of sensors and sources ($M = N$) varying from 6 to 15. For each (M, N), we implement 200 realizations. Three algorithms are considered: `Ono`, `Proposed`, `Proposed` + `refinement`. Given the ground truth $\boldsymbol{R}$, $\boldsymbol{S}$ and the estimates $\tilde{\boldsymbol{R}}$, $\tilde{\boldsymbol{S}}$, the estimation error is calculated as

$$E_{\text{rs}} = \frac{\sum_{i=1}^{M} \|\boldsymbol{r}_i - \tilde{\boldsymbol{r}}_i\| + \sum_{j=1}^{N} \|\boldsymbol{s}_j - \tilde{\boldsymbol{s}}_j\|}{M + N}. \tag{42}$$

We assume that localization accuracy of 0.001 m is enough in practical applications, and thus we lower bound E_{rs} as $E_{\text{rs}} = \max(0.001, E_{\text{rs}})$. In order to account for the intrinsic translational and rotational freedom of the solution, the estimated and true sensor and source positions are optimally aligned by means of Procrustes Analysis [34], which finds the optimal translation and rotation of the solution that minimizes the sum of the distances between each pair of estimated and true positions.

The results are presented in Fig. 7, using the same box-plotting scheme as in Fig. 6. For readability, the estimation error is upper bounded at 3 m in Fig. 7. The overall performance can be ranked as `Ono` < `Proposed` < `Proposed` + `refinement`.

The median value of the estimation errors of `Ono` does not vary significantly with M and N. However, the span of the box increases with increasing M and N. When M and N are large (e.g., > 10), the bottom of the box may possibly reach the lowest estimation bound, i.e., 0.001 m.

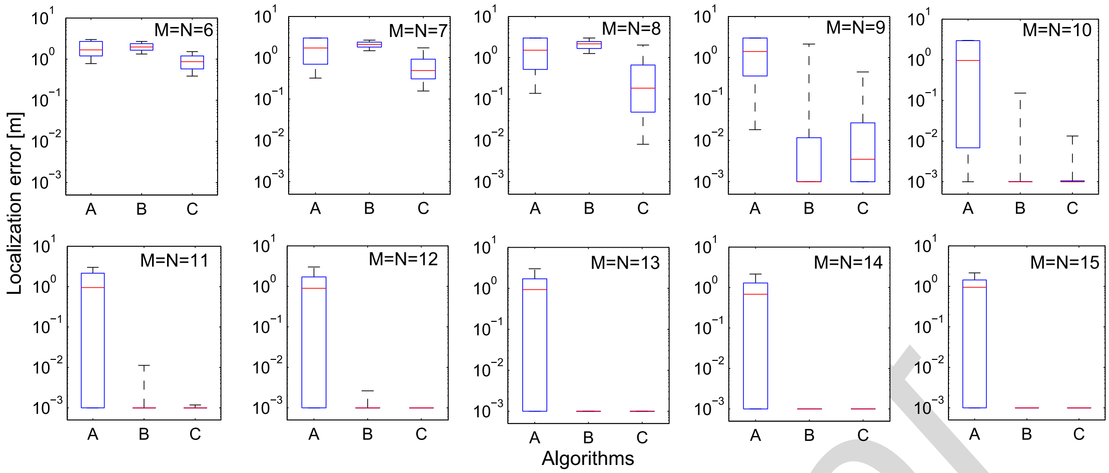


Fig. 7. Localization errors (E_{rs}) for different configurations of M and N: A: Ono; B: Proposed; C: Proposed + refinement.

Proposed performs slightly worse than Ono for $M = N = 8$, where both algorithms achieve large errors. The performance of Proposed improves evidently with increasing M and N. When $M, N \geq 10$, the median estimation error can reach the lowest estimation bound. The span of the box is still wide when $M = N = 10$, but decreases with increasing M and N. When $M, N \geq 13$, the span of the box is close to 0.

Proposed + refinement performs best among all the algorithms. For $M = N = 8$, it outperforms the other two algorithms although all three algorithms achieve large estimation errors. For $M, N \geq 12$, Proposed + refinement achieves a very low estimation error in all cases. For $M, N \geq 13$, Proposed performs similarly to Proposed + refinement.

In summary, Proposed significantly outperforms Ono. Proposed + refinement outperforms Proposed especially when M and N are small.

D. Robustness Analysis

The testing data is generated using equal number of sensors and sources ($M = N$) chosen from [8, 10, 12, 15]. For each (M, N) configuration, we implement 200 realizations. In each realization, the TDOA measurements are corrupted by Gaussian noise with zero mean and a standard deviation $\sigma \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ s. Apart from the considered algorithms, the Cramér-Rao bound (CRB) is calculated as a reference, using the method presented in [13]. The CRB provides a theoretical lower bound on the variance of any estimator, irrespective of the particular estimation method used. We average the estimation errors across 200 realizations:

$$E_{\text{avg}} = 10^{\frac{1}{N_r}\sum_{r=1}^{N_r}\log_{10} E_{\text{rs}}(r)}, \tag{43}$$

where $E_{\text{rs}}(r)$ denotes the estimation error of the r-th realization, $N_r = 200$ is the number of realizations, and the log operation is introduced to reduce the influence of the highly dynamic range of the estimation error. Fig. 8 shows E_{avg} for different TDOA measurement errors (for readability E_{avg} is upper bounded to 3 m). Overall, the performance of the algorithms can still be ranked as Ono < Proposed < Proposed + refinement.

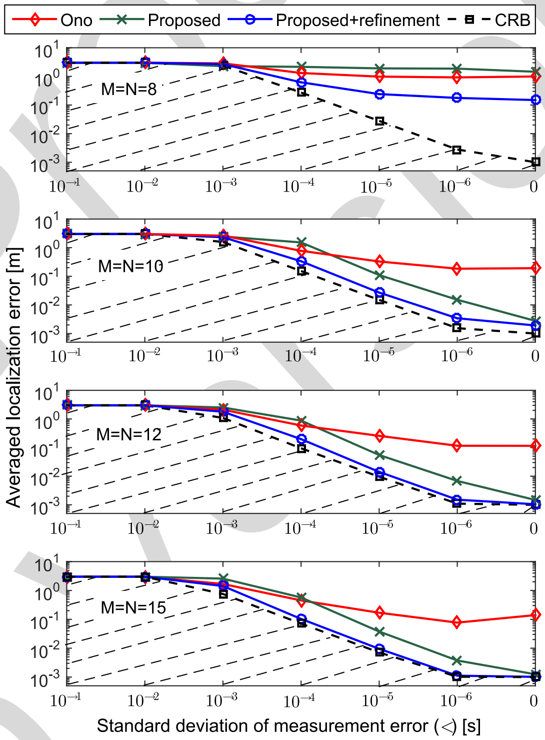


Fig. 8. Average localization errors (E_{avg}) by the considered algorithms in case of TDOA measurement errors. CRB represents the theoretical lower bound of the estimation.

CRB is influenced by M and N only slightly. It mainly depends on the standard deviation of the measurement error. The E_{avg} of CRB decreases when σ decreases. When $\sigma > 10^{-3}$ s, all the algorithms fail. When 10^{-4} s $\leq \sigma \leq 10^{-6}$ s, the performance of all the algorithms improves when σ decreases. Ono generally performs worst among all the algorithms. Proposed performs slightly worse than Ono when $M = N = 8$ but outperforms Ono in other cases. Proposed + refinement performs best. Its performance curve is far from the CRB curve when $M = N = 8$, close to the CRB curve when $M, N \geq 10$, and almost overlapped with the CRB curve when $M = N = 15$. Therefore, Proposed + refinement outperforms Proposed significantly in case of TDOA errors.

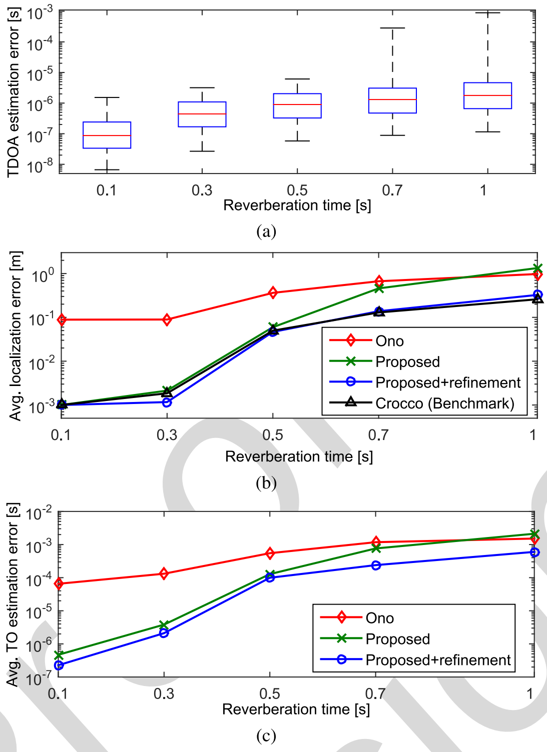
E. Realistic Simulation

We evaluate the performance of the proposed algorithm with realistic TDOA data estimated from simulated ambient sounds. Similarly to previous experiments, we consider a room of size 10 m × 10 m × 3 m, where sensors and sources are randomly distributed with $d_{\max} = 10$ m. The number of sensors and sources are both 15. The impulse responses from the sources to the sensors are simulated by the image-source method [38] with the reverberation time, RT_{60}, controlled by varying the absorption coefficients. The sound speed is 342 m/s and the sampling rate 48 kHz. The sound sources consist of nonoverlapping uttered speech signals, each occupying a time slot of 5 s with the length of speech 3 s. The onset time of each sound source is randomly chosen within the first two seconds of its time slot. The microphone signals are generated by convolving clean speech signals with room impulse responses. The capture time of each microphone is chosen randomly and uniformly from the interval $[-1, 1]$ s. We simulate 5 different reverberant scenarios with reverberation times 0.1 s, 0.3 s, 0.5 s, 0.7 s and 1 s, respectively. For each reverberation time we implement 10 realizations.

The acoustic scenario we consider is a simple case which assumes no overlaps between sound sources and the segmentation of each sound source in the microphone signals can be easily identified. It should be noted that, in practical applications, to find the data association between each sound source and the microphone signals is a nontrivial task, especially when multiple sources are simultaneously active. Although multiple target tracking methods [40], [41] have shown potential to address the data association problem, it still remains an open problem in real-world scenarios.

In this experiment the TDOA of each source with respect to a pair of microphones is estimated from the corresponding segmentations in the microphone signals. A coarse-to-fine scheme [5] is employed to estimate the TDOA, where the two microphone signals are coarsely aligned at first and then processed with the generalized cross-correlation with phase transform (GCC-PHAT) algorithm [39], which is well-known for its robustness to room reverberation. In the GCC-PHAT algorithm, we use a frame length of 8192 with half overlap. As shown in the robustness analysis in Section VII-D, `Proposed` works well when the TDOA estimation error is below 10^{-6} s. In order to achieve such an estimation accuracy, we search in the TDOA space with a small step size of 10^{-9} s.

We use `Ono`, `Proposed` and `Proposed` + `refinement` to estimate the sensor and source locations. For reference, we use the localization results obtained by the Crocco algorithm [13] as `Benchmark`. The Crocco algorithm works on TOA and assumes that both source onset times and device capture times are known. We use the general solution in the Crocco algorithm, which is essentially the same as the geometrical information estimation part of the `Proposed` algorithm. To apply the Crocco algorithm, we derive the TOA from the TDOA (which is estimated from the microphone signals) and the distance between the first microphone and all sources (cf. (1) and (2)), and assume the time offsets between devices to be known. In this way, the estimation errors contained in the estimated TOAs and TDOAs are comparable.



(a)

(b)

(c)

Fig. 9. Performance evaluation for the considered algorithms using TDOAs estimated from ambient sounds in different reverberant scenarios. The numbers of sensors and sources are both 15. (a) TDOA estimation. (b) Sensor and source localization. (c) Time offset (TO) estimation.

We calculate the TDOA estimation errors for all the sensor and source combinations that are contained in the TDOA matrix. The TDOA estimation error is defined as the absolute difference between the estimated value and the ground truth, which is obtained from the prior knowledge of the sensor and source locations and the device capture times. For each reverberation time, we collect all the TDOA estimation errors in 10 realizations and present them in Fig. 9(a), using the same box-plotting scheme as in Fig. 6. It is clearly observed in Fig. 9(a) that the median value of the TDOA estimation errors increases with reverberation time (RT). More outliers (i.e., larger errors) are observed when RT ≥ 0.7 s. For instance, the median value of the errors at RT $= 1$ s is around 10^{-6} s, whereas the upper bound of the error can reach 10^{-3} s.

The average localization errors (E_{avg} in (43)) obtained by the algorithms in different reverberant scenarios are depicted in Fig. 9(b). Overall, the performance degrades with reverberation time, and can be ranked as `Ono` < `Proposed` < `Proposed` + `refinement` ≈ `Benchmark`. The Crocco (`Benchmark`) algorithm, which assumes timing information to be known, also shows localization errors due to large TDOA estimation errors in high reverberation. As shown in Fig. 9(b), `Proposed` can estimate the locations very accurately when RT ≤ 0.3 s. However, its E_{avg} rises to 0.05 m at RT $= 0.5$ s, and becomes larger than 0.5 m when RT ≥ 0.7 s. `Proposed` + `refinement` can significantly improve the localization performance in high reverberation, with E_{avg} being around 0.1 m and 0.2 m for RT $= 0.7$ s and

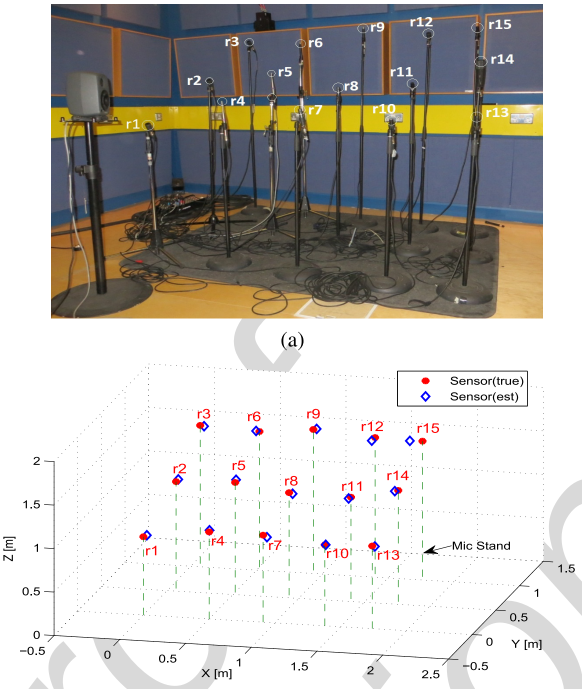
RT $= 1$ s, respectively. `Proposed` + `refinement` can achieve similar performance as `Benchmark`.

With the estimated sensor and source locations, the time offsets between the first microphone and other microphones, as by-products, can be easily calculated using (2). We define the offset time estimation error as the absolute difference between the estimated value and the ground truth, and calculate the average offset time estimation error for 10 realizations, using an equation similar to (43). The results are shown in Fig. 9(c). Since the time offsets are calculated from the sensor and source locations, the performance of time offset estimation depends significantly on sensor and source localization. As shown in Fig. 9(c), the time offset estimation error curves of all the algorithms show similar variation trends to the localization error curves in Fig. 9(b). `Proposed` + `refinement` can achieve time offset estimation errors below one sample (around 20 μs at sampling rate 48 kHz) when RT ≤ 0.3 s. The estimation error rises to around 5 samples (0.1 ms) and 25 samples (0.5 ms) at RT $= 0.5$ s and RT $= 1$ s, respectively.

F. Real Recording

We present an example of using the considered algorithms with real-recorded data in a listening room of size 6 m × 5 m × 3 m and with a reverberation time of around 200 ms. We use 15 microphones of various types such as Sennheiser E600, DPA 4006-TL and Beyerdynamic MCE530, placed as shown in Fig. 10(a). The microphones are connected to an RME Fireface 800 through two Focusrite ISA828 microphone preamplifiers (8 channels each). The sampling rate is 44.1 kHz. The same testing speech used in Section VII-E is played by a Genelec 8010 loudspeaker at 18 different locations (acting as 18 sound sources). To simulate unknown device capture times, a delay randomly chosen from the interval $[-1, 1]$ s is applied to the recording from each microphone. These recordings are used in the localization procedure. We only have the ground truth of the microphone locations, which are manually measured. We investigate microphone localization performance with this ground truth. Although the microphones are placed with a relatively regular topology, in practice the considered algorithms can be used for arbitrary microphone placement.

Compared with the simulated (image-source) scenarios, the performance of TDOA estimation with real recordings may be degraded by extra reflections from microphone stands and also by the fact that the loudspeaker is not a point source. For localization we applied the `Ono`, `Proposed` and `Proposed` + `refinement` algorithms. The average localization errors obtained by the three algorithms are about 0.39 m, 0.03 m and 0.03 m, respectively. The true and estimated microphone locations (`Proposed`) are shown in Fig. 10(b). It is clearly observed that the topology of the microphone network can be recovered. The consistence between the true and estimated microphone locations confirms the potential of the proposed algorithms in real-world applications.



(a)

(b)

Fig. 10. Microphone localization with real-recorded data. (a) Recording environment. (b) Localization result by the `Proposed` algorithm. The average localization error of all the microphones is about 0.03 m.

VIII. Conclusions

We proposed a general solution for sensor and source localization in an ad-hoc array using TDOAs which are biased by unknown time offsets. To overcome the local minima problem when estimating the unknown timing information, we derived the boundary of the timing parameters and utilized a multi-initialization scheme. These strategies can minimize the influence of local minima efficiently. Simulation results demonstrate that the `Proposed` algorithm outperforms existing algorithms in most cases. The performance of the `Proposed` algorithm can be further improved by a refinement strategy. Error analysis demonstrates that the `Proposed` + `refinement` algorithm can get close to the Cramér-Rao bound with at least 10 sensors and sources. Experiments using TDOAs estimated from simulated and real-recorded speech data confirm the potential of the `Proposed` algorithm in real-world applications. Specifically, the `Proposed` + `refinement` algorithm can achieve high-quality estimation for speech data simulated at reverberation time 0.3 s and the estimation error rises to 0.05 m, 0.1 m and 0.2 m for reverberation times 0.5 s, 0.7 s and 1 s, respectively. For real-recorded data in an environment with reverberation time of 0.2 s, the `Proposed` + `refinement` algorithm can achieve an estimation error of 0.03 m.

The `Proposed` algorithm however requires a minimum number of sensors and sources (e.g., $M, N \geq 10$) to guarantee its performance. Although the performance improves significantly as M and N increase, the local minima problem is still not completely solved, e.g., it can still be observed with a low probability even if M and N are large. An interesting direction for future research would be to incorporate prior knowledge of the ad-hoc array into the `Proposed` algorithm, so that the dependence on the number of sensors and sources can be relieved and the local minima problem can be fully overcome. Moreover, although the `Proposed` (+`refinement`) algorithm shows promising results in reverberant scenarios, the robustness to TDOA estimation outliers could be improved by exploiting the redundancy of the TDOA information when using a large number of sensors and sources.

Appendix A
Jacobian Matrix in (19)

For $\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\zeta}_A \\ \lambda\boldsymbol{\zeta}_B \end{bmatrix}$ and $\boldsymbol{\rho} = \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\eta} \\ \boldsymbol{x} \end{bmatrix}$, the Jacobian matrix $\boldsymbol{J} = \frac{\partial\boldsymbol{\zeta}}{\partial\boldsymbol{\rho}}$, which is of size $Q \times (M + N - 1 + 3(N-4))$, can be calculated as

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial\boldsymbol{\zeta}_A}{\partial\boldsymbol{\delta}} & \frac{\partial\boldsymbol{\zeta}_A}{\partial\boldsymbol{\eta}} & \frac{\partial\boldsymbol{\zeta}_A}{\partial\boldsymbol{x}} \\ \frac{\lambda\partial\boldsymbol{\zeta}_B}{\partial\boldsymbol{\delta}} & \frac{\lambda\partial\boldsymbol{\zeta}_B}{\partial\boldsymbol{\eta}} & \frac{\lambda\partial\boldsymbol{\zeta}_B}{\partial\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{J}_{12} & \boldsymbol{J}_{13} \\ \lambda\boldsymbol{J}_{21} & \lambda\boldsymbol{J}_{22} & \lambda\boldsymbol{J}_{23} \end{bmatrix}.$$

The block matrices can be computed as described below.

$$\boldsymbol{J}_{11} = \begin{bmatrix} \frac{\partial[\boldsymbol{U}]_{11}}{\partial\delta_1} & \cdots & \frac{\partial[\boldsymbol{U}]_{11}}{\partial\delta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial\delta_1} & \cdots & \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial\delta_M} \end{bmatrix}.$$

Based on the definition of $\boldsymbol{U}$ in (7), the element-wise differential is expressed as

$$\frac{\partial[\boldsymbol{U}]_{i-1,j-1}}{\partial\delta_k} = \begin{cases} (2\eta_j - 2(t_{1j} - t_{11}))\cdot \uparrow_{i,k}, & k = 1 \\ (2(t_{ij} - t_{i1}) - 2\eta_j)\cdot \uparrow_{i,k}, & k = 2, \ldots, M \end{cases} \tag{A1}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$. Here $\uparrow_{i,k} = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$.

$$\boldsymbol{J}_{12} = \begin{bmatrix} \frac{\partial[\boldsymbol{U}]_{11}}{\partial\eta_1} & \cdots & \frac{\partial[\boldsymbol{U}]_{11}}{\partial\eta_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial\eta_1} & \cdots & \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial\eta_N} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{U}]_{i-1,j-1}}{\partial\eta_k} = -(2(t_{ij} - t_{1j}) + 2\delta_i)\cdot \uparrow_{j,k}, \tag{A2}$$

for $i = 2, \ldots, M$, $j = 2, \ldots, N$, and $k = 2, \ldots, N$.

$$\boldsymbol{J}_{13} = \begin{bmatrix} \frac{\partial[\boldsymbol{U}]_{11}}{\partial[\boldsymbol{X}]_{11}} & \cdots & \frac{\partial[\boldsymbol{U}]_{11}}{\partial[\boldsymbol{X}]_{3,N-4}} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial[\boldsymbol{X}]_{11}} & \cdots & \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial[\boldsymbol{X}]_{3,N-4}} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{U}]_{ij}}{\partial[\boldsymbol{X}]_{kl}} = 0, \tag{A3}$$

for $i = 1, \ldots, M-1$, $j = 1, \ldots, N-1$, $k = 1, \ldots, 3$, and $l = 1, \ldots, N-4$.

Let $\boldsymbol{V} = (\boldsymbol{A} + \boldsymbol{F})\boldsymbol{X} - \boldsymbol{B} - \boldsymbol{G}$.

$$\boldsymbol{J}_{21} = \begin{bmatrix} \frac{\partial[\boldsymbol{V}]_{11}}{\partial\delta_1} & \cdots & \frac{\partial[\boldsymbol{V}]_{11}}{\partial\delta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial\delta_1} & \cdots & \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial\delta_M} \end{bmatrix},$$

where $\frac{\partial[\boldsymbol{V}]_{ij}}{\partial\delta_k} = \frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{ij}}{\partial\delta_k}$.

Since $[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{ij} = \sum_{u=1}^{3}[\boldsymbol{U}]_{iu}[\boldsymbol{X}]_{uj} - [\boldsymbol{U}]_{i,j+3}$, it follows

$$\frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{i-1,j-1}}{\partial\delta_1} = \sum_{u=1}^{3}(2\eta_u - 2(t_{1u} - t_{11}))[\boldsymbol{X}]_{uj} - \{(2\eta_{j+3} - 2(t_{1,j+3} - t_{11})\}, \tag{A4}$$

and

$$\frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{i-1,j-1}}{\partial\delta_k} = \uparrow_{i,k}\cdot \left\{\left(\sum_{u=1}^{3}(2(t_{iu} - t_{i1}) - 2\eta_u)[\boldsymbol{X}]_{uj}\right) + 2\eta_{j+3} - 2(t_{i,j+3} - t_{i1})\right\}, \tag{A5}$$

for $i = 2, \ldots, M$, $j = 2, \ldots, N$, and $k = 2, \ldots, M$.

$$\boldsymbol{J}_{22} = \begin{bmatrix} \frac{\partial[\boldsymbol{V}]_{11}}{\partial\eta_1} & \cdots & \frac{\partial[\boldsymbol{V}]_{11}}{\partial\eta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial\eta_1} & \cdots & \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial\eta_M} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{V}]_{ij}}{\partial\eta_k} = \frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{ij}}{\partial\eta_k}.$$

$$\frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{i-1,j-1}}{\partial\eta_k} = \begin{cases} (-2(t_{ij} - t_{1j}) + 2(\delta_1 - \delta_i))[\boldsymbol{X}]_{kj}, & k = 2, \ldots, 4 \\ (-2(t_{i,j+3} - t_{1,j+3}) + 2(\delta_1 - \delta_i))\cdot \uparrow_{k,j+3}, & k = 5, \ldots, N \end{cases} \tag{A6}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$.

$$\boldsymbol{J}_{23} = \begin{bmatrix} \frac{\partial[\boldsymbol{V}]_{11}}{\partial[\boldsymbol{X}]_{11}} & \cdots & \frac{\partial[\boldsymbol{V}]_{11}}{\partial[\boldsymbol{X}]_{3,N-4}} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial[\boldsymbol{X}]_{11}} & \cdots & \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial[\boldsymbol{X}]_{3,N-4}} \end{bmatrix},$$

where $\frac{\partial[\boldsymbol{V}]_{ij}}{\partial[\boldsymbol{X}]_{kl}} = \frac{\partial[(\boldsymbol{A} + \boldsymbol{F})\boldsymbol{X}]_{ij}}{\partial[\boldsymbol{X}]_{kl}}$.

Since $[(\boldsymbol{A} + \boldsymbol{F})\boldsymbol{X}]_{ij} = \sum_{u=1}^{3}(\boldsymbol{A} + \boldsymbol{F})_{iu}[\boldsymbol{X}]_{uj}$,

$$\frac{\partial[(\boldsymbol{A} + \boldsymbol{F})\boldsymbol{X}]_{ij}}{\partial[\boldsymbol{X}]_{kl}} = \uparrow_{l,j} [\boldsymbol{A} + \boldsymbol{F}]_{ik} \tag{A7}$$

for $i = 1, \ldots, M-1$, $j = 1, \ldots, N-1$, $k = 1, \ldots, 3$, and $l = 1, \ldots, N-4$.

Appendix B
Jacobian Matrix in (39)

For $\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_A \\ \boldsymbol{\nu}_B \end{bmatrix}$ and $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{c} \\ s_{x1} \end{bmatrix}$, the Jacobian matrix $\boldsymbol{I} = \frac{\partial\boldsymbol{\nu}}{\partial\boldsymbol{\beta}}$, which is of size $(M + N - 2) \times 10$, can be calculated as

$$\boldsymbol{I} = \begin{bmatrix} \frac{\partial\boldsymbol{\nu}_A}{\partial\boldsymbol{c}} & \frac{\partial\boldsymbol{\nu}_A}{\partial s_{x1}} \\ \frac{\partial\boldsymbol{\nu}_B}{\partial\boldsymbol{c}} & \frac{\partial\boldsymbol{\nu}_B}{\partial s_{x1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{11} & \boldsymbol{I}_{12} \\ \boldsymbol{I}_{21} & \boldsymbol{I}_{22} \end{bmatrix}.$$

The block matrices can be computed as described below.

Based on the definitions in (30) and (38), it follows

$$[\boldsymbol{\nu}_A]_{i-1} = \sum_{u=1}^{3} [\boldsymbol{D_L C}]_{iu}^2 - 2[\boldsymbol{D_L C}]_{i1} s_{x1} - \bar{g}_i \tag{B1}$$

for $i = 2, \ldots, M$.

$$\boldsymbol{I}_{11} = \begin{bmatrix} \frac{\partial[\boldsymbol{\nu}_A]_1}{\partial[\boldsymbol{C}]_{11}} & \cdots & \frac{\partial[\boldsymbol{\nu}_A]_1}{\partial[\boldsymbol{C}]_{33}} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{\nu}_A]_{M-1}}{\partial[\boldsymbol{C}]_{11}} & \cdots & \frac{\partial[\boldsymbol{\nu}_A]_{M-1}}{\partial[\boldsymbol{C}]_{33}} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{\nu}_A]_{i-1}}{\partial[\boldsymbol{C}]_{kl}} = 2[\boldsymbol{D}_L \boldsymbol{C}]_{il}[\boldsymbol{U}]_{ik} - 2 \uparrow_{l,1} \boldsymbol{U}_{ik} s_{x1} \tag{B2}$$

for $i = 2, \ldots, M$, $k = 1, 2, 3$, and $l = 1, 2, 3$.

$$\boldsymbol{I}_{12} = \begin{bmatrix} \frac{\partial[\boldsymbol{\nu}_A]_1}{\partial s_{x1}} & \cdots & \frac{\partial[\boldsymbol{\nu}_A]_{M-1}}{\partial s_{x1}} \end{bmatrix}^{\mathrm{T}},$$

where

$$\frac{\partial[\boldsymbol{\nu}_A]_i}{\partial s_{x1}} = -2[\boldsymbol{D_L C}]_{i1} \tag{B3}$$

for $i = 1, \ldots, M-1$.

Based on the definitions in (38), it follows

$$[\boldsymbol{\nu}_B]_{j-1} = \sum_{u=1}^{3} [\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3 \boldsymbol{D}_R^{\mathrm{T}}]_{uj}^2 + 2[\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3 \boldsymbol{D}_R^{\mathrm{T}}]_{1j} s_{x1} - \bar{\bar{g}}_j, \tag{B4}$$

for $j = 2, \ldots, N$.

$$\boldsymbol{I}_{21} = \begin{bmatrix} \frac{\partial[\boldsymbol{\nu}_B]_1}{\partial[\boldsymbol{C}]_{11}} & \cdots & \frac{\partial[\boldsymbol{\nu}_B]_1}{\partial[\boldsymbol{C}]_{33}} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{\nu}_B]_{N-1}}{\partial[\boldsymbol{C}]_{11}} & \cdots & \frac{\partial[\boldsymbol{\nu}_B]_{N-1}}{\partial[\boldsymbol{C}]_{33}} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{\nu}_B]_j}{[\boldsymbol{C}]_{kl}} = 2[\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3 \boldsymbol{D}_R^{\mathrm{T}}]_{\cdot,j}^{\mathrm{T}} \frac{\partial \boldsymbol{C}^{-1}}{\partial[\boldsymbol{C}]_{kl}} [\boldsymbol{\Sigma}_3 \boldsymbol{D}_R^{\mathrm{T}}]_{\cdot,j} + 2 s_{x1} \frac{\partial \boldsymbol{C}^{-1}}{\partial[\boldsymbol{C}]_{kl}} [\boldsymbol{\Sigma}_3 \boldsymbol{D}_R^{\mathrm{T}}]_{\cdot,j}, \tag{B5}$$

for $j = 1, \ldots, N-1$, $k = 1, 2, 3$, and $l = 1, 2, 3$. Using the inversion theory, we have

$$\frac{\partial \boldsymbol{C}^{-1}}{\partial[\boldsymbol{C}]_{kl}} = -\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial[\boldsymbol{C}]_{kl}} \boldsymbol{C}^{-1} = -\boldsymbol{C}^{-1} \boldsymbol{O}_{kl} \boldsymbol{C}^{-1}, \tag{B6}$$

where $\boldsymbol{O}_{kl}$ is a matrix with the same size of $\boldsymbol{C}$, with the (k, l)-th element being 1 and other elements being 0.

$$\boldsymbol{I}_{22} = \begin{bmatrix} \frac{\partial[\boldsymbol{\nu}_B]_1}{\partial s_{x1}} & \cdots & \frac{\partial[\boldsymbol{\nu}_B]_{N-1}}{\partial s_{x1}} \end{bmatrix}^{\mathrm{T}},$$

where, for $j = 1, \ldots, N-1$,

$$\frac{\partial[\boldsymbol{\nu}_B]_j}{\partial s_{x1}} = 2[\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3 \boldsymbol{D}_R^{\mathrm{T}}]_{j1}. \tag{B7}$$

REFERENCES

[1] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *Proc. IEEE Symp. Communications and Vehicular Technol. in the Benelux*, Ghent, Belgium, 2011, pp. 1–6.

[2] M. H. Hennecke and G. A. Fink, "Towards acoustic self-localization of ad hoc smartphone arrays," in *Proc. IEEE Joint Workshop on Hands-Free Speech Communication and Microphone Arrays*, Edinburgh, UK, 2011, pp. 127–132.

[3] L. Wang, "Multi-band multi-centroid clustering based permutation alignment for frequency-domain blind speech separation," *Digital Signal Process.*, vol. 31, pp. 79–92, Aug. 2014.

[4] L. Wang, T. Gerkmann, and S. Doclo, "Noise power spectral density estimation using MaxNSR blocking matrix," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 9, pp. 493–1508, Sep. 2015.

[5] N. Ono, H. Kohno, N. Ito, and S. Sagayama, "Blind alignment of asynchronously recorded signals for distributed microphone array," in *Proc. IEEE Workshop on Applicat. of Signal Process. to Audio and Acoustics*, New York, USA, 2009, pp. 161–164.

[6] S. T. Birchfield and A. Subramanya, "Microphone array position calibration by basis-point classical multidimensional scaling," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1025–1034, Sep. 2005.

[7] S. Wilson, J. Walters, and J. Abel, "Speaker locations from inter-speaker range measurements: Closed-form estimator and performance relative to the Cramer-Rao lower bound," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Montreal, Canada, 2004, pp. 389–392.

[8] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. Int. Conf. Embedded Networked Sensor Systems*, Baltimore, USA, 2004, pp. 50–61.

[9] M. J. Taghizadeh, R. Parhizkar, P. N. Garner, H. Bourlard, and A. Asaei, "Ad hoc microphone array calibration: Euclidean distance matrix completion algorithm and theoretical guarantees," *Signal Process.*, vol. 107, pp. 123–140, Feb. 2015.

[10] K. K. Chintalapudi, A. Dhariwal, R. Govindan, and G. Sukhatme, "Ad-hoc localization using ranging and sectoring," in *Proc. IEEE Infocom*, Hong Kong, China, 2004, pp. 2662–2672.

[11] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: A high accuracy acoustic ranging system using COTS mobile devices," in *Proc. Int. Conf. Embedded Networked Sensor Systems*, Sydney, Australia, 2007, pp. 1–14.

[12] J. Herrera and H. S. Kim, "Ping-pong: Using smartphones to measure distances and relative positions," in *Proc. Meetings on Acoustics*, San Francisco, USA, 2013, pp. 1–10.

[13] M. Crocco, A. Del Bue, and V. Murino, "A bilinear approach to the position self-calibration of multiple sensors," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 660–673, Feb. 2012.

[14] V. C. Raykar, I. V. Kozintsev, and R. Lienhart, "Position calibration of microphones and loudspeakers in distributed computing platforms," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 70–83, Jan. 2005.

[15] S. Thrun, "Affine structure from sound," in *Advances in Neural Information Processing System*, Y. Weiss, B. Scholkopf, and J. Platt, Eds. Cambridge, USA: MIT Press, 2005, pp. 1353–1360.

[16] R. Heusdens and N. D. Gaubitch, "Time-delay estimation for TOA-based localization of multiple sensors," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Florence, Italy, 2014, pp. 609–613.

[17] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, "Auto-localization in ad-hoc microphone arrays," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Vancouver, Canada, 2013, pp. 106–110.

[18] F. Jiang, Y. Kuang, and K. Astrom, "Time delay estimation for TDOA self-calibration using truncated nuclear norm regularization," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Vancouver, Canada, 2013, pp. 3885–3889.

[19] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, "Calibration of distributed sound acquisition systems using TOA measurements from a moving acoustic source," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Florence, Italy, 2014, pp. 7455–7459.

[20] I. McCowan, M. Lincoln, and I. Himawan, "Microphone array shape calibration in diffuse noise fields," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 3, pp. 666–670, Mar. 2008.

[21] M. J. Taghizadeh, P. N. Garner, and H. Bourlard, "Enhanced diffuse field model for ad hoc microphone array calibration," *Signal Process.*, vol. 101, pp. 242–255, Aug. 2014.

[22] P. Pertila, M. S. Hamalainen, and M. Mieskolainen, "Passive temporal offset estimation of multichannel recordings of an ad-hoc microphone array," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 11, pp. 2393–2402, Nov. 2013.

[23] V. Deleskog, H. Habberstad, G. Hendeby, D. Lindgren, and N. Wahlstrom, "Robust NLS sensor localization using MDS initialization," in *Proc. Int. Conf. Information Fusion*, Salamanca, Spain, 2014, pp. 1–7.

[24] M. Parviainen, P. Pertila, and M. S. Hamalainen, "Self-localization of wireless acoustic sensors in meeting rooms," in *Proc. Joint Workshop Hands-Free Speech Communication and Microphone Arrays*, Nancy, France, 2014, pp. 152–156.

[25] T. K. Hon, L. Wang, J. D. Reiss, and A. Cavallaro, "Audio fingerprinting for multi-device self-localization," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 10, pp. 1623–1636, Oct. 2015.

[26] R. Biswas and S. Thrun, "A passive approach to sensor network localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 1544–1549.

[27] M. Chen, Z. Liu, L. W. He, P. Chou, and Z. Zhang, "Energy-based position estimation of microphones and speakers for ad hoc microphone arrays," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, USA, 2007, pp. 22–25.

[28] Z. Liu, Z. Zhang, L. W. He, and P. Chou, "Energy-based sound source localization and gain normalization for ad hoc microphone arrays," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Honolulu, USA, 2007, pp. 761–764.

[29] X. Sheng and Y. H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, Jan. 2005.

[30] M. Pollefeys and D. Nister, "Direct computation of sound and microphone locations from time-difference-of-arrival data," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Las Vegas, USA, 2008, pp. 2445–2448.

[31] Y. Kuang, S. Burgess, A. Torstensson, and K. Astrom, "A complete characterization and solution to the microphone position self-calibration problem," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Vancouver, Canada, 2013, pp. 3875–3879.

[32] Y. Kuang and K. Astrom, "Stratified sensor network self-calibration from TDOA measurements," in *Proc. European Signal Process. Conf.*, Marrakech, Morroco, 2013, pp. 1–5.

[33] S. Zhayida, F. Andersson, Y. Kuang, and K. Astrom, "An automatic system for microphone self-localization using ambient sound," in *Proc. European Signal Process. Conf.*, Lisbon, Portugal, 2014, pp. 954–958.

[34] G. A. F. Seber, *Multivariate Observations*. Hoboken, USA: John Wiley & Sons Inc., 1984.

[35] J. Nocedal and S. Wright, *Numerical Optimization*. New York, USA: Springer, 1999.

[36] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. Annual ACM Symp. Theory of Computing*, Palo Alto, USA, 2013, pp. 665–674.

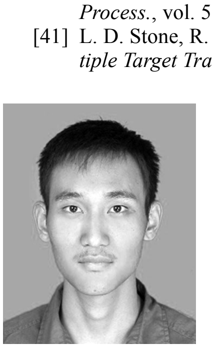
[37] S. Van Huffel, H. Park, and J. B. Rosen, "Formulation and solution of structured total least norm problems for parameter estimation," *IEEE Trans. Signal Process.*, vol. 44, no. 10, pp. 2464–2474, Oct. 1996.

[38] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, Apr. 1979.

[39] M. S. Brandstein and H. F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Munich, Germany, 1997, pp. 375–378.

[40] C. Hue, J. P. Le Cadre, and P. Perez, "Sequential Monte Carlo methods for multiple target tracking and data fusion," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 309–325, Feb. 2002.

[41] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian Multiple Target Tracking*. Boston, USA: Artech House, 2013.



Lin Wang received the B.S. degree in electronic engineering from Tianjin University, China, in 2003; and the Ph.D degree in signal processing from Dalian University of Technology, China, in 2010. From 2011 to 2013, he was an Alexander von Humboldt Fellow at the University of Oldenburg, Germany. Since 2014, he has been a postdoctoral researcher in the Centre for Intelligent Sensing at Queen Mary University of London. His research interests include video and audio compression, microphone array, blind source separation, and 3D audio processing.

Tsz-Kin Hon received his B.Eng. degree in electronic and computer engineering from the Hong Kong University of Science and Technology (HKUST) in 2006; and the Ph.D. degree in digital signal processing from Kings College London (KCL) in 2013. He was a Research Engineer in the R&D of the Giant Electronic Ltd. between 2006 and 2009. He is currently a Postdoctoral Research Assistant in the Centre for Intelligent Sensing at Queen Mary University of London. His research interests include audio and video signal processing, device localization and synchronization, multi-source signal processing, joint time–frequency analysis and filtering, acoustic echo cancellation, speech enhancement, and biomedical signal processing.

Joshua D. Reiss is a Reader in Audio Engineering with the Centre for Digital Music in the School of Electronic Engineering and Computer Science at Queen Mary University of London. He has bachelors degrees in both physics and mathematics, and earned his Ph.D. in physics from the Georgia Institute of Technology. He is a member of the Board of Governors of the Audio Engineering Society, and co-founder of the company MixGenius, now known as LandR. Dr. Reiss has published more than 100 scientific papers and serves on several steering and technical committees. He has investigated sound synthesis, time scaling and pitch shifting, source separation, polyphonic music transcription, loudspeaker design, automatic mixing for live sound, and digital audio effects. His primary focus of research, which ties together many of the above topics, is on the use of state-of-the-art signal processing techniques for professional sound engineering.

Andrea Cavallaro received the Ph.D. degree in electrical engineering from Swiss Federal Institute of Technology, Lausanne, Switzerland, in 2002. He was a Research Fellow with British Telecommunications in 2004. He is a Professor of Multimedia Signal Processing and the Director of the Centre for Intelligent Sensing at Queen Mary University of London. He has authored more than 150 journal and conference papers, one monograph on Video Tracking (Wiley, 2011), and three edited books, Multi-Camera Networks (Elsevier, 2009), Analysis, Retrieval and Delivery of Multimedia Content (Springer, 2012), and Intelligent Multimedia Surveillance (Springer, 2013). Prof. Cavallaro is an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING and a member of the editorial board of the IEEE MultiMedia Magazine. He is an elected member of the IEEE Image, Video, and Multidimensional Signal Processing Technical Committee, and is the Chair of its Awards Committee, and an elected member of the IEEE Circuits and Systems Society Visual Communications and Signal Processing Technical Committee. He served as an elected member of the IEEE Signal Processing Society Multimedia Signal Processing Technical Committee, as an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA and IEEE TRANSACTIONS ON SIGNAL PROCESSING, as an Associate Editor and as an Area Editor of IEEE Signal Processing Magazine, and as a Guest Editor of eleven special issues of international journals. He was General Chair for the IEEE/ACM ICDSC 2009, BMVC 2009, M2SFA2 2008, SSPE 2007, and IEEE AVSS 2007. He was Technical Program Chair of the IEEE AVSS 2011, the European Signal Processing Conference in 2008, and WIAMIS 2010. He received the Royal Academy of Engineering Teaching Prize in 2007, three Student Paper Awards on target tracking and perceptually sensitive coding at the IEEE ICASSP in 2005, 2007, and 2009, respectively, and the Best Paper Award at IEEE AVSS 2009.

Self-Localization of Ad-Hoc Arrays Using Time Difference of Arrivals

Lin Wang, Tsz-Kin Hon, Joshua D. Reiss, and Andrea Cavallaro

***Abstract*—We investigate the problem of sensor and source joint localization using time-difference of arrivals (TDOAs) of an ad-hoc array. A major challenge is that the TDOAs contain unknown time offsets between asynchronous sensors. To address this problem, we propose a low-rank approximation method that does not need any prior knowledge of sensor and source locations or timing information. At first, we construct a pseudo time of arrival (TOA) matrix by introducing two sets of unknown timing parameters (source onset times and device capture times) into the current TDOA matrix. Then we propose a Gauss-Newton low-rank approximation algorithm to jointly identify the two sets of unknown timing parameters, exploiting the low-rank property embedded in the pseudo TOA matrix. We derive the boundaries of the timing parameters to reduce the initialization space and employ a multi-initialization scheme. Finally, we use the estimated timing parameters to correct the pseudo TOA matrix, which is further applied to sensor and source localization. Experimental results show that the proposed approach outperforms state-of-the-art algorithms.**

***Index Terms*—Ad-hoc array, low rank approximation, self-localization, time-difference of arrival.**

I. Introduction

AD-HOC sensor networks composed of randomly distributed and independent recording devices, such as smartphones, wireless microphones, and laptops, have been attracting increased interest due to their flexibility in sensor placement [1], [2]. However, the geometrical configuration of an ad-hoc array is generally unknown and may change with time. Device localization is a very important task in this context. Although not necessary in blind source separation [3] and adaptive beamforming [4], a precise knowledge of the device locations is still required in many applications such as fixed beamforming, source localization and tracking [5]. A straightforward approach for device localization is to measure the distance among all the device pairs and to apply closed-form estimators to recover their spatial locations [6]–[9]. The inter-device distance can be directly measured using tapes or laser pointers, which is time consuming [10]. The inter-device distance can also be computed based on the acoustical transfer delays between two devices, by actively communicating calibration sounds between independent devices [11], [12]. Active communication requires a specially designed network interface or software, which is not available in many recording devices. Passive device localization instead exploits external acoustic events emitted from discrete spatial positions. Two types of information, time of arrival (TOA) and time difference of arrival (TDOA), can be estimated from microphone recordings and be used to jointly localize sensors and sources [13], [14]. If the source signal is known beforehand (predefined sound), its TOA for each individual microphone is obtained through cross-correlation with the given signal. If the source signal is unknown (ambient sound), the TDOA for a pair of microphones is estimated through cross-correlation of the two microphone signals.

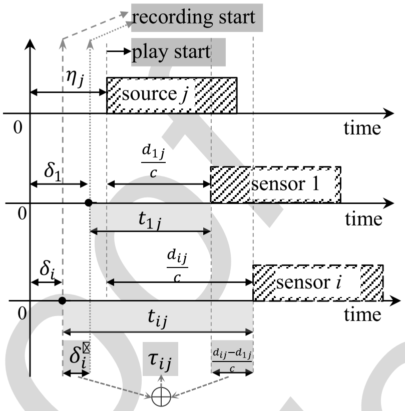


Fig. 1. Illustration of TOA (t_{1j} and t_{ij}) and TDOA (τ_{ij}) for source j and two sensors 1 and i. The source has unknown onset time η_j. The sensors have unknown capture start times δ_1 and δ_i, and unknown time offset δ_i'.

A challenge that arises for an ad-hoc array is that its devices are usually not synchronized. As a result, practical TOA measurements are biased because they may include an unknown source onset time and an unknown device capture time (Fig. 1). Similarly, practical TDOA measurements may include an unknown time offset between a pair of recordings. These time uncertainties make the sensor localization problem harder [14], [17].

Manuscript received May 16, 2015; revised August 21, 2015; revised August 21, 2015; accepted October 11, 2015. This work was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/K007491/1.

L. Wang is with Centre for Intelligent Sensing, Queen Mary University of London, London, E1 4NS, UK (e-mail: lin.wang@qmul.ac.uk; Phone: +44 20 7882 7549).

T.-K. Hon is with Centre for Intelligent Sensing, Queen Mary University of London, London, E1 4NS, UK (e-mail: tsz.kin.hon@qmul.ac.uk, thomas_htk@hotmail.com; Phone: +44 20 7882 7549).

J. D. Reiss is with Centre for Intelligent Sensing, Queen Mary University of London, London, E1 4NS, UK (e-mail: joshua.reiss@qmul.ac.uk; Phone: +44 20 7882 7982).

A. Cavallaro is with Centre for Intelligent Sensing, Queen Mary University of London, London, E1 4NS, UK (e-mail: a.cavallaro@qmul.ac.uk; Phone: +44 20 7882 5165).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2498130

To address the unknown timing information, one can estimate all the unknown parameters, including sensor and source locations and time offsets, simultaneously [5]. The former approach usually leads to the minimization of a non-convex cost function. To overcome local minima problems, some approaches introduce additional constraints, such as co-located sensor-source pairs [14] or far-field sound sources [15]. However, these constraints are not always satisfied in practical applications. Alternatively, one can estimate the unknown timing information, and then estimate the sensor and source locations [19]. This simplifies the problem by decomposing it into two independent stages. The low-rank structure embedded in the TOA measurements can be exploited to estimate the unknown onset times and capture times [16]–[18]. These approaches typically involve a gradient-based optimization procedure, which is sensitive to local minima. Moreover, existing algorithms usually simplify the optimization by assuming either the onset times or the capture times to be known. To the best of our knowledge, a general but effective solution is still missing.

In this paper we present a method that jointly estimates the sensor and source locations from the biased TDOA measurements. We employ the two-stage estimation framework, i.e., timing information estimation followed by sensor and source localization. When estimating the timing information, the existing low-rank approximation techniques are not applicable to TDOA because the latter does not contain the desired low-rank structure as TOA does. To address this challenge, we construct a pseudo TOA matrix by introducing two sets of unknown timing parameters (source onset times and device capture times) into the current TDOA matrix. To estimate these two parameters from the pseudo TOA, we propose a Gauss-Newton low-rank approximation algorithm.

The main novelty of the proposed algorithm is summarized as follows. The proposed complete solution is in contrast to the state-of-the-art timing information estimation algorithms [16]–[18], which assume that only one set of parameters (either onset times or capture times) are unknown. The proposed algorithm has several features which are helpful to tackle local minima problems. First, it is shown that the onset times and capture times in the pseudo TOA matrix can be represented as functions of the sensor and source locations, and particularly, these functions are independent of the TOA data. This feature makes it possible to derive the boundary of the parameter space, and to reduce the initialization space with a reasonable assumption of the maximum sensor-source distance. Second, the algorithm can converge quickly to either local or global minima, and typically shows evident distinction between the two minima. This feature makes it both computationally and technically feasible to employ a multi-initialization scheme to increase the probability of reaching a global minimum.

The paper is organized as follows. Section II overviews related work of self-localization. Section III formulates the problem and introduces existing solutions. The proposed method, including pseudo TDOA matrix construction, timing information estimation, and geometrical information estimation, is described in Sections IV–VI, respectively. Performance evaluation is conducted in Section VII and conclusions are drawn in Section VIII.

II. Related Work

Passive device self-localization approaches can be divided into two classes, namely pairwise distance estimation and joint sensor and source localization (Table I).

Pairwise distance estimation approaches recover the distance between each pair of devices and then the relative locations of all the devices. One can assume a diffuse noise field and estimate the microphone pairwise distance by fitting the measured noise coherence to the theoretical model of the field [20], [21]. Although this solution alleviates the need for activating several sources, it is only feasible for small arrays, and requires the microphones to be synchronized in order to calculate the coherence correctly. Moreover, the diffuse noise assumption is not always satisfied. Another approach is to determine the inter-device distance by estimating the minimum and maximum TDOA values [22]–[25]. By assuming that the minimum and maximum TDOAs come from sources located at the end-fire directions, the inter-device distance can be calculated without knowing the time offset. However, the assumption of end-fire sources is not always satisfied.

Joint localization approaches estimate the locations of sensors and sources simultaneously using acoustic events emitted from various spatial locations. Information that can be used for the joint localization task includes TOA, TDOA, and received signal energy. The TOA measurement may be biased by the unknown source onset time and device capture time, while the TDOA measurement may be biased by the unknown time offset. To minimize the influence of unknown timing information, two schemes can be employed: one-stage estimation [5], [14] and two-stage estimation [16], [17].

The *one-stage* scheme estimates the unknown position and timing information simultaneously via a maximum likelihood estimation, using either TOA or TDOA information [5], [14], [26]. The method presented in [26] jointly estimates the locations and onset times, utilizing the TOAs of calibration signals. The presented method in [14] jointly estimates the locations and capture times from the TOAs or TDOAs of calibration signals. A closed-form solution is also derived by assuming pairs of co-located sensors and sources, e.g., in laptops or tablets [14]. The method presented in [5] jointly estimates the locations and capture times from the TDOAs measured from ambient sounds. An auxiliary function-based iterative algorithm, which shows a better convergence property, is proposed for the optimization procedure. If the timing information is already known, the one-stage estimation can be performed by exploiting the affine geometry of the sensors and sources [13], [15]. A singular value decomposition (SVD) based low-rank matrix factorization approach is employed to reduce the number of unknown parameters, which are further estimated through a much simpler non-linear optimization procedure. The method in [15] estimates the locations of the sensors and sources from the TOAs of the sources, which are assumed to be at the far field and hence have the same onset time. This method is further extended to a general case which does not rely on the far-field assumption [13], [30]. A closed-form solution can be derived by assuming a pair of co-located sensor and source [13]. Besides TDOA and

TABLE I
SUMMARY OF DEVICE SELF-LOCALIZATION ALGORITHMS. (DT: DEVICE CAPTURE TIME; ST: SOURCE START TIME; DL: DEVICE LOCATION; SL: SOURCE LOCATION; PD: PAIRWISE DISTANCE).

Reference	Acoustic Signal	Information Used	Approach		Constraint	Parameters Needed	Parameters Estimated
[20], [21]	ambient sound	mic signal coherence	pairwise distance + close-form estimator	noise coherence match	diffuse noise field, small array	DT	PD
[22]–[25]	ambient sound	TDOA		Min/Max TDOA estimation	end-fire sources	–	PD
[5]	ambient sound	TDOA	joint sensor/source localization	one-stage estimation	–	–	DT, DL, SL
[14]					co-located sensor/source	–	DT, DL, SL
[15]					far-field sources	DT	DL, SL
[13]	calibration signal	TOA			–	DT, ST	DL, SL
[26]					–	DT	ST, DL, SL
[14]					co-located sensor/source	ST	DT, DL, SL
[27], [28]	ambient sound	mic signal energy			co-located sensor/source	–	DL, SL
[16]	calibration signal	TOA		two-stage estimation	–	ST	DT, DL, SL
[17]–[19] [30]–[33]	ambient sound				–	DT	ST, DL, SL
proposed	ambient sound	TDOA			–	–	DT, DL, SL

TOA, the energy of the microphone signals can also be exploited for sensor and source localization, using a maximum likelihood optimization procedure [27]–[29]. The energy-based method does not need exact timing information, but usually assumes pairs of co-located sensors and sources, e.g., in laptops [27], [28]. In summary, most one-stage estimation algorithms assume that at least one set of timing information is known (either onset time or capture time) or impose additional geometrical constraints to simplify the optimization problem. Only [5] presents a general formulation.

A *two-stage* scheme that first estimates the timing information is more suitable for TOA measurements. After TOA correction, the scheme jointly estimates the sensor and source locations utilizing similar techniques to those employed in the one-stage scheme, e.g., [5], [13], [14]. An effective way of estimating timing information is to exploit the low-rank structure of the sensor and source locations, which all lie in the 3-dimensional space [30]. Specifically, a matrix consisting of the biased TOA measurements can be formulated which is supposed to be rank-3 if the bias is correctly compensated. With these constraints, the timing information can be estimated so that the mentioned matrix becomes rank-3 after compensation. Three algorithms have been proposed recently to solve this low-rank approximation problem, including alternating minimization [17], [19], nuclear truncation minimization [18], [31], [32], and structured total least squares [16]. The structured total least squares algorithm converges more than 100 times faster than the other two algorithms [16]. The two-stage estimation scheme is promising for the localization problem since by decomposing the problem into two stages, the optimization becomes easier.

In summary, the above-mentioned three low-rank approximation algorithms are suitable for TOA only, and assume that at least one timing information to be known (either onset time or capture time). For now, no progress has been reported on applying these algorithms to biased TDOA measurements. The proposed algorithm will fill this gap.

III. PROBLEM FORMULATION

We denote vectors and matrices in bold lowercase letters and bold uppercase letters, respectively. The operator $[\cdot]_i$ denotes the i-th element of a vector while $[\cdot]_{ij}$ denotes the (i,j)-th element of a matrix. The operator vec_c denotes vectorization of a matrix by concatenating its column vectors, while the operator vec_r denotes vectorization of a matrix by concatenating its row vectors.

A. Signal Model

Consider M independent acoustic sensors (microphones) and N sound sources (speakers) at unknown locations $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_M]_{3\times M}$ and $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N]_{3\times N}$, respectively. The emission onset times of the sources are unknown and represented as $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_N]^{\mathrm{T}}$, where the superscript $(\cdot)^{\mathrm{T}}$ denotes transpose. The unknown capture start times of the sensors are $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_M]^{\mathrm{T}}$. The TOA from the j-th source to the i-th sensor is defined as

$$t_{ij} = \frac{\|\boldsymbol{r}_i - \boldsymbol{s}_j\|}{c} + \eta_j - \delta_i, \tag{1}$$

where c is the speed of sound and $\|\cdot\|$ is the Euclidean norm. The TOAs for the combination of all the sensors and sources are represented as a matrix $\boldsymbol{T} = [t_{ij}]$ where $i = 1, \ldots, N$ and $j = 1, \ldots, M$. The TDOA from the j-th source to the first and the i-th sensor is defined as

$$\tau_{ij} = t_{ij} - t_{1j} = \frac{\|\boldsymbol{r}_i - \boldsymbol{s}_j\| - \|\boldsymbol{r}_1 - \boldsymbol{s}_j\|}{c} + \delta_1 - \delta_i, \tag{2}$$

where $\delta'_i = \delta_1 - \delta_i$ is the time offset between the first and the i-th sensor, and $d_{ij} = \|\boldsymbol{r}_i - \boldsymbol{s}_j\|$ is the distance between the i-th sensor and the j-th source (Fig. 1). The TDOAs for the combination of all the sensors and sources is represented as $\boldsymbol{T}_e = [\tau_{ij}]$ where $i = 1, \ldots, N$ and $j = 1, \ldots, M$. Based on (2), the first row of $\boldsymbol{T}_e$ consists of all zero elements, i.e., $\tau_{1,\cdot} = 0$. The time offsets of all the sensors with respect to the first sensor is $\boldsymbol{\delta}' = [\delta'_1, \ldots, \delta'_M]$, where $\delta'_1 = 0$.

We assume the sound sources are unknown (uncontrolled ambient sounds) and thus only TDOA information is available, which is biased by the unknown time offset. The goal of this paper is to estimate the sensor and source locations $\boldsymbol{R}$ and $\boldsymbol{S}$ from the biased TDOA measurements $\boldsymbol{T}_e$. Since the TDOA contains only the distance information, the estimation of the locations is not invariant against rotation, translation and reflection. Without loss of generality, we assume that the first sensor lies at the spatial origin $\boldsymbol{r}_1 = [0,0,0]^{\mathrm{T}}$ while the first source is confined to lie on the first positive axis, i.e., $\boldsymbol{s}_1 = [s_{x1}, 0, 0]^{\mathrm{T}}$ with $s_{x1} > 0$. We also assume that the onset time of the first source, η_1, is 0.

B. Existing Solutions

1) One-Stage Estimation: The sensor and source locations and the time offsets can be simultaneously estimated via maximum likelihood estimation. This is expressed as

$$\begin{aligned}&\hat{\boldsymbol{R}}, \hat{\boldsymbol{S}}, \hat{\boldsymbol{\delta}}' \\ &= \arg \min_{\boldsymbol{R},\boldsymbol{S},\boldsymbol{\delta}'} \sum_{i=2}^{M} \sum_{j=1}^{N} \left(\frac{\|\boldsymbol{r}_i - \boldsymbol{s}_j\| - \|\boldsymbol{r}_1 - \boldsymbol{s}_j\|}{c} + \delta'_i - \tau_{ij} \right)^2 .\end{aligned} \tag{3}$$

A gradient-based solution has been presented in [5]. However, with many unknown parameters, the minimization of (3) leads to a non-convex problem, which is sensitive to parameter initialization and can easily get stuck in local minima.

2) Two-Stage Estimation: The low-rank structure embedded in the TOA measurements can be exploited to estimate the timing and location information separately [17].

From (1) it can be derived that

$$\frac{\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i + \boldsymbol{s}_j^{\mathrm{T}}\boldsymbol{s}_j - 2\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{s}_j}{c^2} = t_{ij}^2 + \eta_j^2 + \delta_i^2 - 2(t_{ij}\eta_j - t_{ij}\delta_i + \eta_j\delta_i), \tag{4}$$

for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. Using the assumption $\eta_1 = 0$ and sequentially subtracting the corresponding equation for $i = 1$ and the equation for $j = 1$ from the general form (4), it follows

$$\begin{aligned}&\frac{-2(\boldsymbol{r}_i - \boldsymbol{r}_1)^{\mathrm{T}}(\boldsymbol{s}_j - \boldsymbol{s}_1)}{c^2} \\ &= t_{ij}^2 - t_{i1}^2 - t_{1j}^2 + t_{11}^2 + 2\delta_i(t_{ij} - t_{i1}) - 2\delta_1(t_{1j} - t_{11}) \\ &\quad - 2\eta_j(t_{ij} - t_{1j}) + 2\eta_j(\delta_1 - \delta_i)\end{aligned} \tag{5}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$. Equation (5) can be expressed in a matrix form as

$$\frac{-2\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}}}{c^2} = \boldsymbol{D} + \boldsymbol{U}. \tag{6}$$

The elements of $\bar{\boldsymbol{R}}_{3\times(M-1)}$, $\bar{\boldsymbol{S}}_{3\times(N-1)}$, $\boldsymbol{D}_{(M-1)\times(N-1)}$, and $\boldsymbol{U}_{(M-1)\times(N-1)}$ can be expressed as

$$\begin{aligned}\bar{\boldsymbol{R}}_{\cdot i-1} &= \boldsymbol{r}_i - \boldsymbol{r}_1, \\ \bar{\boldsymbol{S}}_{\cdot j-1} &= \boldsymbol{s}_j - \boldsymbol{s}_1, \\ \boldsymbol{D}_{i-1,j-1} &= t_{ij}^2 - t_{i1}^2 - t_{1j}^2 + t_{11}^2, \\ \boldsymbol{U}_{i-1,j-1} &= 2\delta_i(t_{ij} - t_{i1}) - 2\delta_1(t_{1j} - t_{11}) \\ &\quad - 2\eta_j(t_{ij} - t_{1j}) + 2\eta_j(\delta_1 - \delta_i),\end{aligned} \tag{7}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$.

Based on the above definitions, the matrices $\bar{\boldsymbol{R}}$ and $\bar{\boldsymbol{S}}$ are related to the sensor and source locations and their matrix ranks are both equal to 3. Thus, the rank of the left side of (6), $\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}}$, is equal to 3 and the same is true for the right side, $\boldsymbol{D} + \boldsymbol{U}$. This low-rank information can be utilized to estimate both the timing and geometrical information.

The matrix $\boldsymbol{D}$ is constructed from the TOA parameters t_{ij}, while the matrix $\boldsymbol{U}$ is constructed from the unknown parameters $\boldsymbol{\eta}$ and $\boldsymbol{\delta}$. Timing information estimation is essentially a low-rank matrix completion problem [36], i.e., the matrix $\boldsymbol{U}$ can be seen as a modification of $\boldsymbol{D}$ so that $\hat{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{U}$ is of rank 3 [17]. Low-rank approximation algorithms, such as alternative minimization [17], nuclear truncation minimization [18] and structured total least square (STLS) [16], have been proposed to solve this problem. After estimating the timing information and correcting the matrix, $\hat{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{U}$, the rank-3 of $\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}}$ is further used to estimate $\boldsymbol{R}$ and $\boldsymbol{S}$.

However, this solution is not directly applicable to our problem, where only TDOA is available and both onset times and capture times are unknown. The TDOA does not contain the desired low-rank structure as in (6). Existing low-rank approximation algorithms [16]–[18] assume either the onset times or the capture times to be known. To address these challenges, we propose a sensor and source localization method, which is described in Sections IV–VI.

IV. Pseudo TOA Matrix Construction

To employ the low-rank approximation method, a TOA matrix is needed which contains a low-rank structure as in (6). Since we only have the TDOA matrix $\boldsymbol{T}_e$ available, one possible solution is to convert $\boldsymbol{T}_e$ to a matrix containing the desired low-rank structure. To this end, we define a pseudo TOA matrix $\dot{\boldsymbol{T}}$ in the same form as $\boldsymbol{T}$ in (1), i.e.,

$$\dot{t}_{ij} = \frac{\|\boldsymbol{r}_i - \boldsymbol{s}_j\|}{c} + \dot{\eta}_j - \dot{\delta}_i, \tag{8}$$

where $\dot{\eta}_j$ and $\dot{\delta}_i$ are two unknown parameters termed as *pseudo onset time* and *pseudo capture time*, respectively. If we can find appropriate $\dot{\eta}_j$ and $\dot{\delta}_i$ that enable $\dot{\boldsymbol{T}} = \boldsymbol{T}_e$, the TDOA matrix would be interpreted as a TOA matrix.

To make $\dot{\boldsymbol{T}} = \boldsymbol{T}_e$, it requires $\dot{t}_{ij} = \tau_{ij}, \forall i, j$ or, more specifically,

$$\begin{bmatrix} \dot{t}_{11} & \cdots & \dot{t}_{1N} \\ \dot{t}_{21} & \cdots & \dot{t}_{2N} \\ \vdots & \ddots & \vdots \\ \dot{t}_{M1} & \cdots & \dot{t}_{MN} \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \\ \tau_{21} & \cdots & \tau_{2N} \\ \vdots & \ddots & \vdots \\ \tau_{M1} & \cdots & \tau_{MN} \end{bmatrix}. \tag{9}$$

Using (2) and (8) on $\dot{t}_{ij} = \tau_{ij}$, it follows that

$$\dot{\eta}_j - \dot{\delta}_i = -\frac{\|\boldsymbol{r}_1 - \boldsymbol{s}_j\|}{c} + \delta'_i, \tag{10}$$

for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. It can be easily verified that the two sets of pseudo parameters $\dot{\eta}_j$ ($j = 1, \ldots, N$) and $\dot{\delta}_i$ ($i = 1, \ldots, M$) can be uniquely determined by solving a group of (10) if the sensor locations $\boldsymbol{R}$, the source locations $\boldsymbol{S}$ and the time offsets $\boldsymbol{\delta}'$ are known. This demonstrates theoretically that there do exist appropriate $\dot{\eta}_j$ and $\dot{\delta}_i$ that enable $\dot{\boldsymbol{T}} = \boldsymbol{T}_e$. In practice, these two parameters are unknown and will have to be estimated from $\boldsymbol{T}_e$.

Thus, by introducing two pseudo parameters, the TDOA matrix $\boldsymbol{T}_e$ can be interpreted as the pseudo TOA matrix $\dot{\boldsymbol{T}}$. Obviously, $\dot{\boldsymbol{T}}$ and its pseudo parameters $\dot{\eta}_j$ and $\dot{\delta}_i$ are different from the original TOA, onset time and capture time, which are defined in (1). However, with similar structure as the original TOA matrix, $\dot{\boldsymbol{T}}$ can be exploited for low-rank approximation. The next task will be to estimate $\dot{\boldsymbol{\delta}} = [\dot{\delta}_1, \ldots, \dot{\delta}_M]^{\mathrm{T}}$ and $\dot{\boldsymbol{\eta}} = [\dot{\eta}_2, \ldots, \dot{\eta}_N]^{\mathrm{T}}$ and to recover $\boldsymbol{R}$ and $\boldsymbol{S}$.

Considering that the concepts of original TOA, onset time and capture time will not be used hereafter, we remove the accent $(\dot{\ })$ of $\dot{\boldsymbol{T}}$, $\dot{\boldsymbol{\delta}}$ and $\dot{\boldsymbol{\eta}}$, and also call them TOA, onset time and capture time, without introducing ambiguities.

V. Timing Information Estimation

In this section, we propose a Gauss-Newton low-rank approximation algorithm to estimate the unknown parameters $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ from the pseudo TOA matrix $\boldsymbol{T}$.

As shown in Section III-B2, the timing information estimation can be seen as a low-rank matrix completion problem (6). We solve this problem under a structured total least-square framework [16]. The matrices $\boldsymbol{D}, \boldsymbol{U}$, and $\hat{\boldsymbol{D}}$ are each partitioned into two blocks, such that the first submatrix contains the first three columns while the second submatrix contains the remaining columns. This is expressed as

$$\boldsymbol{D} = [\boldsymbol{A}_{M-1\times 3} \,|\, \boldsymbol{B}_{M-1\times N-4}], \quad (11)$$

$$\boldsymbol{U} = [\boldsymbol{F}_{M-1\times 3} \,|\, \boldsymbol{G}_{M-1\times N-4}], \quad (12)$$

$$\hat{\boldsymbol{D}} = [(\boldsymbol{A}+\boldsymbol{F}) \,|\, (\boldsymbol{B}+\boldsymbol{G})], \quad (13)$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{F}$ and $\boldsymbol{G}$ are the partitioned submatrices. The objective is to find an appropriate $\boldsymbol{U}$ (consisting of $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$) and an additional matrix $\boldsymbol{X}$ so that

$$(\boldsymbol{A}+\boldsymbol{F})\boldsymbol{X} = \boldsymbol{B}+\boldsymbol{G}. \quad (14)$$

If (14) holds, the matrix $\boldsymbol{B}+\boldsymbol{G}$ will lie in the range of the matrix $\boldsymbol{A}+\boldsymbol{F}$. It follows that

$$\operatorname{rank}([(\boldsymbol{A}+\boldsymbol{F}) \,|\, (\boldsymbol{B}+\boldsymbol{G})]) = \operatorname{rank}(\boldsymbol{A}+\boldsymbol{F}) = 3. \quad (15)$$

Similarly to [16], the STLS cost function is defined as

$$\boxed{\xi(\boldsymbol{\delta}, \boldsymbol{\eta}, \boldsymbol{X}) = \|\boldsymbol{U}\|_F^2 + \lambda^2 \|(\boldsymbol{A}+\boldsymbol{F})\boldsymbol{X} - \boldsymbol{B} - \boldsymbol{G}\|_F^2} \quad (16)$$

where $\|\cdot\|_F$ denotes the F-norm. The first term on the right side of (16) is introduced to prevent algorithm divergence; the second term corresponds to the condition (14) and is emphasized with a large penalty λ. In [16] a simplified solution is given, which however assumes that the onset time $\boldsymbol{\eta}$ is known. This solution is not feasible for our case, where both $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ are unknown. We present a complete and general solution below.

Let us define the unknown parameters $\boldsymbol{\delta}$, $\boldsymbol{\eta}$, and $\boldsymbol{X}$ as a column vector $\boldsymbol{\rho} = [\boldsymbol{\delta}^{\mathrm{T}} \quad \boldsymbol{\eta}^{\mathrm{T}} \quad \boldsymbol{x}^{\mathrm{T}}]^{\mathrm{T}}$, where $\boldsymbol{x} = \operatorname{vec}_c(\boldsymbol{X}) = [[\boldsymbol{X}]_{11}, [\boldsymbol{X}]_{21}, \ldots, [\boldsymbol{X}]_{12}, [\boldsymbol{X}]_{22}, \ldots]^{\mathrm{T}}$. The cost function (16) can be further written as

$$\begin{aligned}\xi(\boldsymbol{\rho}) &= \sum_{i=1}^{M-1}\sum_{j=1}^{N-1}[\boldsymbol{U}]_{ij}^2 + \lambda^2 \sum_{i=1}^{M-1}\sum_{j=1}^{N-4}[(\boldsymbol{A}+\boldsymbol{F})\boldsymbol{X} - \boldsymbol{B} - \boldsymbol{G}]_{ij}^2 \\ &= \sum_{q=1}^{Q} \zeta_q^2, \end{aligned} \quad (17)$$

where $Q = (M-1)(N-1) + (M-1)(N-4)$ and $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_A^{\mathrm{T}} \quad \lambda\boldsymbol{\zeta}_B^{\mathrm{T}}]^{\mathrm{T}} = [\zeta_1, \ldots, \zeta_Q]^{\mathrm{T}}$, with the two column vectors

$$\begin{cases} \boldsymbol{\zeta}_A = \operatorname{vec}_c(\boldsymbol{U}) \\ \boldsymbol{\zeta}_B = \operatorname{vec}_c((\boldsymbol{A}+\boldsymbol{F})\boldsymbol{X} - \boldsymbol{B} - \boldsymbol{G}) \end{cases}. \quad (18)$$

Based on (17), the minimization of ξ is a nonlinear least square problem, which can be solved with the well-known Gauss-Newton algorithm [35]. Suppose the Jacobian matrix

$$\boldsymbol{J} = \frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\rho}} \quad (19)$$

can be obtained. The unknown $\boldsymbol{\rho}$ can be estimated in an iterative way as

$$\boldsymbol{\rho}^{(m+1)} = \boldsymbol{\rho}^{(m)} - ({\boldsymbol{J}^{(m)}}^{\mathrm{T}} \boldsymbol{J}^{(m)})^{-1} {\boldsymbol{J}^{(m)}}^{\mathrm{T}} \boldsymbol{\zeta}^{(m)}, \quad (20)$$

where $\boldsymbol{\rho}^{(m)}$, $\boldsymbol{J}^{(m)}$ and $\boldsymbol{\zeta}^{(m)}$ denote the parameters obtained in the m-th iteration. The computation of the Jacobian matrix is presented in Appendix A.

To tackle the local minima problem, we reduce the initialization space by computing the boundary of the parameters (Section V-A). We also use multiple initializations to increase the probability of reaching the global minimum (Section V-B).

A. Parameter Boundary

We derive the boundary of the unknown parameters by using the definitions of $\boldsymbol{T}$ and t_{ij} in (8)–(10), and the assumptions $\boldsymbol{r}_1 = [0 \quad 0 \quad 0]^{\mathrm{T}}$ and $\eta_1 = 0$.

Using $t_{11} = 0$, it follows that

$$\frac{d_{11}}{c} + \delta_1 = 0 \Rightarrow \delta_1 = -\frac{\|\boldsymbol{s}_1\|}{c}. \quad (21)$$

Using $t_{1j} = 0$ for $j = 2, \ldots, N$, it follows that

$$\frac{d_{11}}{c} + \delta_1 + \eta_j = 0 \Rightarrow \eta_j = \frac{\|\boldsymbol{s}_1\| - \|\boldsymbol{s}_j\|}{c} \quad (22)$$

$$\Rightarrow -\frac{\|\boldsymbol{s}_j\|}{c} < \eta_j < \frac{\|\boldsymbol{s}_1\|}{c}. \quad (23)$$

Averaging on the i-th ($i = 2, \ldots, M$) row of $\boldsymbol{T}$, it follows

$$\frac{1}{N}\sum_{j=1}^{N} t_{ij} = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{d_{ij}}{c} + \delta_i + \eta_j\right) \quad (24)$$

$$\Rightarrow \delta_i = \frac{1}{N}\sum_{j=1}^{N} t_{ij} - \frac{1}{N}\sum_{j=1}^{N}\left(\frac{d_{ij}}{c} + \eta_j\right). \quad (25)$$

Using (22), it follows that

$$\sum_{j=1}^{N}\left(\frac{d_{ij}}{c}+\eta_j\right)=\frac{d_{i1}}{c}+\sum_{j=2}^{N}\left(\frac{d_{ij}+\|\boldsymbol{s}_1\|-\|\boldsymbol{s}_j\|}{c}\right). \quad (26)$$

Using (26) in (25), it follows that

$$\frac{1}{N}\left(\sum_{j=1}^{N}t_{ij}-\frac{d_{i1}}{c}-\sum_{j=2}^{N}\left(\frac{d_{ij}+\|\boldsymbol{s}_1\|}{c}\right)\right) < \delta_i < \frac{1}{N}\left(\sum_{j=1}^{N}t_{ij}+\sum_{j=2}^{N}\left(\frac{\|\boldsymbol{s}_j\|}{c}\right)\right). \quad (27)$$

Suppose the maximum distance between a sensor and a source is known to be $d_{\max}$. Using $d_{\max}$ in (21), (23) and (27), the boundary of the parameters can be obtained as

$$\boxed{\begin{cases} -\frac{d_{\max}}{c}<\eta_j<\frac{d_{\max}}{c} \\ \frac{1}{N}\sum\limits_{j=1}^{N}t_{ij}-\frac{d_{\max}}{c}<\delta_1<\frac{1}{N}\sum\limits_{j=1}^{N}t_{ij}+\frac{d_{\max}}{c} \\ \frac{1}{N}\sum\limits_{j=1}^{N}t_{ij}-\frac{(2N-1)d_{\max}}{Nc}<\delta_i<\frac{1}{N}\sum\limits_{j=1}^{N}t_{ij}+\frac{(N-1)d_{\max}}{cN} \end{cases}} \quad (28)$$

This is an interesting result since, based on (28), the values of the onset times and capture times are only related to the locations of sensors and sources (more specifically $d_{\max}$) and a constant term $\frac{1}{N}\sum_{j=1}^{N}t_{ij}$. With (28), the initialization space can be significantly reduced compared to (unconstrained) random initialization. For example, assuming $d_{\max}=10$ m, $c=342$ m/s, and $N=10$, the parameter space can be initialized as $\delta_1-\frac{1}{N}\sum_{j=1}^{N}t_{ij}\in[-0.03,0.03]$ s, $\delta_i-\frac{1}{N}\sum_{j=1}^{N}t_{ij}\in[-0.06,0.03]$ s, and $\eta_j\in[-0.03,0.03]$ s.

B. Multiple Initializations

To overcome the local minima problem we use multiple initializations, a scheme that is suitable for the low-rank approximation algorithm. First, the Gauss-Newton algorithm can converge at an approximately quadratic rate, which takes only a few iterations to reach global or local minima [35]. This makes it computationally feasible to try multiple initializations. Second, we experimentally observe that the difference between the global and local minima is very evident, making it easy to detect whether a global minimum has been reached.

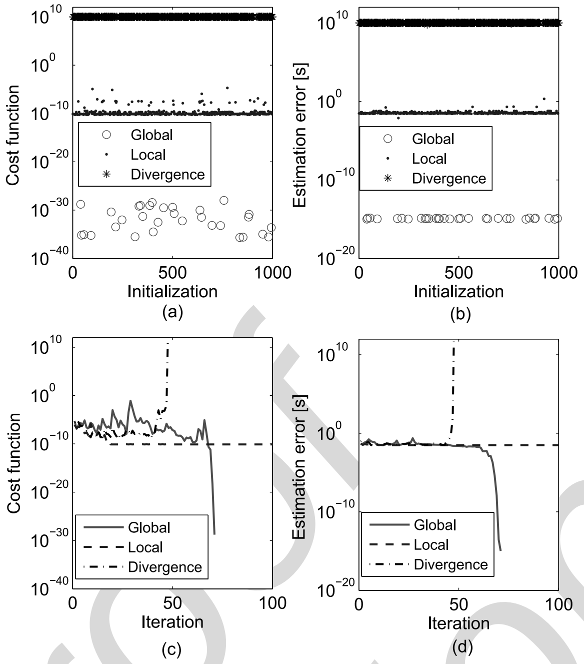


Fig. 2. Convergence results in terms of (a) cost function and (b) estimation error by the Gauss-Newton low-rank approximation algorithm with 1000 initializations ($M=N=10$). Typical iteration curves of (c) cost function and (d) estimation error for global converge, local convergence, and divergence.

Fig. 2 gives an example of the convergence behaviour of the Gauss-Newton low-rank approximation algorithm, using 10 sources and 10 sensors randomly distributed with $d_{\max}=10$ m. The start times and capture times are chosen randomly and uniformly from the range $[-1,1]$ s. The algorithm uses 1000 random initializations uniformly chosen from the range defined by (28). The results are shown in terms of cost function $\|\boldsymbol{\xi}_B\|_F^2$, where $\boldsymbol{\xi}_B$ is defined in (18), and estimation error ($\|\boldsymbol{\rho}-\tilde{\boldsymbol{\rho}}\|$ for the true value $\boldsymbol{\rho}$ and the estimated value $\tilde{\boldsymbol{\rho}}$). Fig. 2(a) depicts the final cost function obtained by each initialization. Three types of results can be achieved: global convergence, local convergence, and divergence. Their difference can be clearly observed: the cost function values below 10^{-20} in case of global convergence, around 10^{-10} in case of local convergence, and over 10^{10} in case of divergence. This makes it easy to detect the convergence status of the algorithm. Accordingly, Fig. 2(b) depicts the final estimation error by each initialization. The three types of convergence result in different estimation errors: these errors are around 10^{-15} s in case of global convergence, around 0.1 s in case of local convergence and over 10^{10} s in case of divergence. Fig. 2(c) and (d) depict typical cost function and estimation error curves during the Gauss-Newton iteration. With quadric convergence rate, the algorithm can converge either locally or globally within 100 iterations.

The description of the timing information estimation algorithm is given in Algorithm 1. The input $\boldsymbol{T}$ equals the TDOA $\boldsymbol{T}_e$. We try multiple initializations until the algorithm reaches global convergence. For each initialization, $\boldsymbol{\delta}^{(0)}$ and $\boldsymbol{\eta}^{(0)}$ are randomly chosen within the range (28).

The value of the cost function threshold (ζ_{T2} in Algorithm 1) is chosen based on the convergence results obtained with clean TDOA measurements. This threshold can be easily determined since the final cost functions for local and global convergence show evident differences (e.g., we choose a threshold of 10^{-15} according to Fig. 2(c)). For noisy inputs, the final cost function for global convergence increases with TDOA errors, making the difference between local and global convergence less evident. Thus, the optimal threshold may vary with the amount of noise. However, we will use the same threshold for all scenarios. There are two reasons for that. First, since the final cost function for global convergence increases with TDOA errors, the algorithm will reach global convergence once the cost function is below the threshold, which is determined for clean TDOA measurements. Second, if in case of noisy inputs the final cost function for global convergence is higher than the threshold, the algorithm will run until reaching the maximum number of initializations and choose the solution with the minimum cost function. This multi-initialization scheme can reduce the influence of a too-low threshold.

Algorithm 1 Gauss-Newton low-rank approximation for timing information estimation

```
Input: $\boldsymbol{T}$
Output: $\boldsymbol{\rho}_o = \{\boldsymbol{\delta}, \boldsymbol{\eta}, \boldsymbol{X}\}$
Parameters: $\zeta_{T1}$: threshold for divergence
            $\zeta_{T2}$: threshold for global convergence
            $d_{\rho T}$: stopping threshold for Gauss-Newton iterations
            $m_{T1}$: maximum number of initializations
            $m_{T2}$: maximum number of iterations
            $\lambda$: penalty constant in (16)
            $d_{\max}$: maximum sensor-source distance in (28)
$f_1 = 0; m_1 = 0; \zeta_o = \mathrm{Inf};$
while $f_1 == 0$ do
    Initialize $\boldsymbol{\delta}^{(0)}, \boldsymbol{\eta}^{(0)}$; (28)
    Compute $\boldsymbol{D}, \boldsymbol{A}, \boldsymbol{B}$; (7)(11)
    $\boldsymbol{X}^{(0)} = (\boldsymbol{A}+\boldsymbol{F})^{-1}(\boldsymbol{B}+\boldsymbol{G})$; $f_2 = 0$; $m_2 = 0$;
    while $f_2 == 0$ do
        Compute $\boldsymbol{U}^{(m_2)}, \boldsymbol{F}^{(m_2)}, \boldsymbol{G}^{(m_2)}$; (7)
        Compute $\boldsymbol{\zeta}^{(m_2)} = \begin{bmatrix} \boldsymbol{\zeta}_A^{(m_2)} \\ \lambda\boldsymbol{\zeta}_B^{(m_2)} \end{bmatrix}^{\mathrm{T}}$; (17)
        Compute $\boldsymbol{J}^{(m_2)}$; (19)
        Update $\boldsymbol{\rho}^{(m_2+1)}$; (20)
        if $\|\boldsymbol{\zeta}_B^{(m_2)}\|_F^2 > \zeta_{T1}$ OR $\|\boldsymbol{\Delta\rho}\| < d_{\rho T}$ OR
        $m_2 == m_{T2}$
            then $f_2 = 1$;
        end if
        $m_2 \leftarrow m_2 + 1$;
    end while
    if $\|\boldsymbol{\zeta}_B^{(m_2-1)}\|_F^2 < \zeta_o$ then
        $\zeta_o = \|\boldsymbol{\zeta}_B^{(m_2-1)}\|_F^2$; $\boldsymbol{\rho}_o = \boldsymbol{\rho}^{(m_2-1)}$;
    end if
    if $\zeta_o < \zeta_{T2}$ OR $m_1 == m_{T1}$; then $f_1 = 1$; end if
    $m_1 \leftarrow m_1 + 1$;
end while
```

VI. Geometrical Information Estimation

Building on the work in [13], [17], in this section we present a Gauss-Newton joint localization algorithm to estimate the sensor and source locations, using the timing information obtained in Section V.

Given the estimated $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$, (6) becomes $\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}} = -\frac{c^2}{2}\hat{\boldsymbol{D}}$. Since $\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}}$ is of rank 3, it can be decomposed via singular value decomposition (SVD) as

$$\bar{\boldsymbol{R}}^{\mathrm{T}}\bar{\boldsymbol{S}} = \boldsymbol{D}_L\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}}, \tag{29}$$

where $\boldsymbol{\Sigma}_3$ keeps only the three largest singular values while $\boldsymbol{D}_L$ and $\boldsymbol{D}_R$ consist of the corresponding left-singular and right-singular vectors, respectively. With (29), $\bar{\boldsymbol{R}}$ and $\bar{\boldsymbol{S}}$ can be calculated as

$$\begin{cases} \bar{\boldsymbol{R}}^{T} = \boldsymbol{D}_L\boldsymbol{C} \\ \bar{\boldsymbol{S}} = \boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}} \end{cases}, \tag{30}$$

where $\boldsymbol{C}$ is an arbitrary 3×3 matrix which is invertible. Using the assumptions $\boldsymbol{r}_1 = [0,0,0]^{\mathrm{T}}$, $\boldsymbol{s}_1 = [s_{x1},0,0]^{\mathrm{T}}$, the sensor and source locations can be calculated as

$$\begin{cases} \boldsymbol{r}_i = \bar{\boldsymbol{r}}_i \\ \boldsymbol{s}_j = \bar{\boldsymbol{s}}_j + [\, s_{x1} \quad 0 \quad 0 \,]^{\mathrm{T}} \end{cases}. \tag{31}$$

For localization, we only need to estimate 10 parameters regardless of the number of sensors and sources. The 10 unknown parameters are represented as $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{c} \\ s_{x1} \end{bmatrix}$ where $\boldsymbol{c} = \mathrm{vec}_c(\boldsymbol{C}) = [[\boldsymbol{C}]_{11}, [\boldsymbol{C}]_{21}, \ldots, [\boldsymbol{C}]_{33}]^{\mathrm{T}}$.

To estimate $\boldsymbol{\beta}$, we rewrite the relationship (4) as

$$\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i + \boldsymbol{s}_j^{\mathrm{T}}\boldsymbol{s}_j - 2\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{s}_j = g_{ij}, \tag{32}$$

with $g_{ij} = c^2(t_{ij}^2 + \eta_j^2 + \delta_i^2 - 2(t_{ij}\eta_j - t_{ij}\delta_i + \eta_j\delta_i))$. We further have

$$\begin{cases} \boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i - \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{r}_1 - 2(\boldsymbol{r}_i^{\mathrm{T}} - \boldsymbol{r}_1^{\mathrm{T}})\boldsymbol{s}_j = g_{ij} - g_{1j} \\ \boldsymbol{s}_j^{\mathrm{T}}\boldsymbol{s}_j - \boldsymbol{s}_1^{\mathrm{T}}\boldsymbol{s}_1 - 2\boldsymbol{r}_i^{\mathrm{T}}(\boldsymbol{s}_j - \boldsymbol{s}_1) = g_{ij} - g_{i1} \end{cases}. \tag{33}$$

Using (31) in (33), it follows that

$$\begin{cases} \bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{r}}_i - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}(\bar{\boldsymbol{s}}_j + \boldsymbol{s}_1) = g_{ij} - g_{1j} \\ \bar{\boldsymbol{s}}_j^{\mathrm{T}}\bar{\boldsymbol{s}}_j + 2\bar{\boldsymbol{s}}_j^{\mathrm{T}}\boldsymbol{s}_1 - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{s}}_j = g_{ij} - g_{i1} \end{cases}. \tag{34}$$

Using (6) in (34), it follows that

$$\boxed{\begin{cases} \bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{r}}_i - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}\boldsymbol{s}_1 = \bar{g}_i \\ \bar{\boldsymbol{s}}_j^{\mathrm{T}}\bar{\boldsymbol{s}}_j + 2\bar{\boldsymbol{s}}_j^{\mathrm{T}}\boldsymbol{s}_1 = \bar{\bar{g}}_j \end{cases}} \tag{35}$$

where

$$\begin{cases} \bar{g}_{ij} = g_{ij} - g_{1j} - c^2[\hat{\boldsymbol{D}}]_{ij} \\ \bar{\bar{g}}_{ij} = g_{ij} - g_{i1} - c^2[\hat{\boldsymbol{D}}]_{ij} \end{cases}, \tag{36}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$. It can be easily verified that $\bar{g}_{ij}$ has the same value for any j while $\bar{\bar{g}}_{ij}$ has the same value for any i. Thus they can be represented as $\bar{g}_i$ and $\bar{\bar{g}}_j$.

The $M+N-2$ equations in (35) can be used to estimate $\boldsymbol{\beta}$. The cost function is defined as

$$\begin{aligned}\varsigma(\boldsymbol{\beta}) &= \sum_{i=2}^{M}(\bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{r}}_i - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}\boldsymbol{s}_1 - \bar{g}_i)^2 + \sum_{j=2}^{N}(\bar{\boldsymbol{s}}_j^{\mathrm{T}}\bar{\boldsymbol{s}}_j + 2\bar{\boldsymbol{s}}_j^{\mathrm{T}}\boldsymbol{s}_1 - \bar{\bar{g}}_j)^2 \\ &= \sum_{p=1}^{M+N-2}\nu_p^2, \end{aligned} \tag{37}$$

where $\boldsymbol{\nu} = [\boldsymbol{\nu}_A^{\mathrm{T}} \quad \boldsymbol{\nu}_B^{\mathrm{T}}]^{\mathrm{T}} = [\nu_1, \ldots, \nu_{M+N-2}]^{\mathrm{T}}$, with

$$\begin{cases} [\boldsymbol{\nu}_A]_{i-1} = \bar{\boldsymbol{r}}_i^{\mathrm{T}}\bar{\boldsymbol{r}}_i - 2\bar{\boldsymbol{r}}_i^{\mathrm{T}}\boldsymbol{s}_1 - \bar{g}_i \\ [\boldsymbol{\nu}_B]_{j-1} = \bar{\boldsymbol{s}}_j^{\mathrm{T}}\bar{\boldsymbol{s}}_j + 2\bar{\boldsymbol{s}}_j^{\mathrm{T}}\boldsymbol{s}_1 - \bar{\bar{g}}_j \end{cases}, \tag{38}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$.

The minimization of ς is a nonlinear least square problem which can be solved with the Gauss-Newton algorithm. Suppose the Jacobian matrix

$$\boldsymbol{I} = \frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\beta}} \tag{39}$$

can be obtained. $\boldsymbol{\beta}$ can be estimated iteratively as

$$\boldsymbol{\beta}^{(m+1)} = \boldsymbol{\beta}^{(m)} - (\boldsymbol{I}^{(m)^{\mathrm{T}}}\boldsymbol{I}^{(m)})^{-1}\boldsymbol{I}^{(m)^{\mathrm{T}}}\boldsymbol{\nu}^{(m)} \tag{40}$$

where $\boldsymbol{I}^{(m)}$, $\boldsymbol{\beta}^{(m)}$ and $\boldsymbol{\nu}^{(m)}$ denote the parameters obtained in the m-th iteration. The detailed derivation of the Jacobian matrix is given in Appendix B.

Compared with [13], [17], which only consider $\boldsymbol{\nu}_A$, the proposed algorithm considers both $\boldsymbol{\nu}_A$ and $\boldsymbol{\nu}_B$, i.e., more observations are used. This may lead to better convergence behaviour. Similarly to Algorithm 1, we use a multi-initialization scheme. The description of the Gauss-Newton joint localization algorithm is given in Algorithm 2. The elements in $\boldsymbol{\beta}^{(0)}$ are initialized as normally distributed random numbers.

The convergence behaviour of the Gauss-Newton joint localization algorithm is closely related to the accuracy of timing information estimation. If the low-rank approximation algorithm only converges locally, (29) does not exactly hold and the joint localization algorithm converges locally or even diverges. If the low-rank approximation algorithm converges globally with accurate timing information estimation, the joint localization algorithm can globally converge easily with a multi-initialization scheme. Fig. 3 gives an example of sensor and source localization using the same data as Fig. 2. In Fig. 3(a), the low-rank approximation algorithm converges locally and so does the joint localization algorithm. In Fig. 3(b), the low-rank approximation algorithm converges globally and the joint localization algorithm can recover the sensor and source locations accurately.

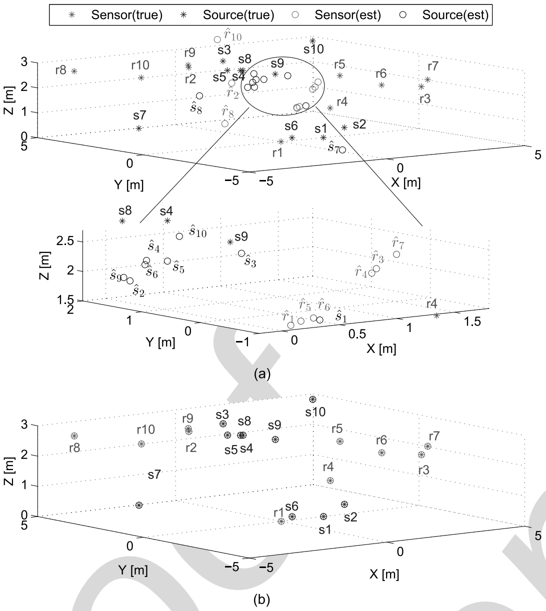


Fig. 3. Sensor and source localization by the proposed two-stage method ($M = N = 10$). The Gauss-Newton low-rank approximation algorithm and the Gauss-Newton joint localization algorithm both converge (a) locally and (b) globally.

Algorithm 2 Gauss-Newton joint localization algorithm for geometrical information localization

```
Input: $\boldsymbol{T}$, $\boldsymbol{\delta}$, $\boldsymbol{\eta}$
Output: $\boldsymbol{R}$, $\boldsymbol{S}$, $\boldsymbol{\beta}_o = \{\boldsymbol{C}, s_{x1}\}$
Parameters: $\nu_{T1}$: threshold for divergence
        $\nu_{T2}$: threshold for global convergence
        $d_{\beta T}$: stopping threshold for Gauss-Newton iterations
        $m_{T3}$: maximum number of initializations
        $m_{T4}$: maximum number of iterations
Compute $\hat{\boldsymbol{D}}$; (13)
Compute $\boldsymbol{D}_L$, $\boldsymbol{D}_R$, $\boldsymbol{\Sigma}_3$ via SVD; (29)
$f_3 = 0$; $m_3 = 0$; $\nu_o = \mathrm{Inf}$;
while $f_3 == 0$ do
    Initialize $\boldsymbol{\beta}^{(0)}$; $f_4 = 0$; $m_4 = 0$;
    while $f_4 == 0$ do
        Compute $\bar{\boldsymbol{R}}^{(m_4)}$, $\bar{\boldsymbol{S}}^{(m_4)}$; (30)
        Compute $\boldsymbol{\nu}^{(m_4)}$, $\boldsymbol{I}^{(m_4)}$; (38)(39)
        Update $\boldsymbol{\beta}^{(m_4+1)}$; (40)
        if $\|\boldsymbol{\nu}^{(m_4)}\|_F^2 > \nu_{T1}$ OR $\|\boldsymbol{\Delta\beta}\| < d_{\beta T}$ OR $m_4 == m_{T4}$
            then $f_4 = 1$;
        end if
        $m_4 \leftarrow m_4 + 1$;
    end while
    if $\|\boldsymbol{\nu}^{(m_4-1)}\|_F^2 < \nu_o$ then
        $\nu_o = \|\boldsymbol{\nu}(m_4 - 1)\|_F^2$; $\boldsymbol{\beta}_o = \boldsymbol{\beta}^{(m_4-1)}$;
    end if
    if $\nu_o < \nu_{T2}$ OR $m_3 == m_{T3}$; then $f_3 = 1$; end if
    $m_3 \leftarrow m_3 + 1$;
end while
Compute $\boldsymbol{R}$, $\boldsymbol{S}$; (31)
```

VII. Experimental Results

A. Experiment Setup

The proposed algorithm consists of two stages: the Gauss-Newton low-rank approximation algorithm for timing information estimation and the Gauss-Newton joint localization algorithm for geometrical information estimation. The first sub-algorithm is sensitive to local minima. Five experiments are thus carried out: the first experiment examines the convergence property of the timing information estimation algorithm; the second experiment compares the performance of the proposed algorithm with existing algorithms; the third investigates the robustness of the proposed algorithm to noisy TDOA measurements; the fourth evaluates the performance with realistic TDOA data estimated from simulated ambient sounds; the last one presents an example with real-recorded data.

Monte Carlo simulations are used to generate testing data. We test all the possible combinations of different numbers of sensors (M) and sources (N), where M is chosen from [5:15, 20] and N is chosen from [5:15, 20, 25]. For each (M, N), we implement different realizations. In each realization, the locations of the sensors and sources are uniformly distributed inside an enclosure of 10 m × 10 m × 3 m with $d_{\max} = 10$ m, whereas the onset times and capture times are chosen randomly and uniformly from the range $[-1, 1]$ s.

A fair comparison of the proposed algorithm with existing algorithms is problematic since most existing approaches assume some prior knowledge of the timing or location information. The closest approach to ours is the one proposed by Ono [5] which does not consider any prior knowledge. Three algorithms are therefore considered: the one-stage algorithm (`Ono`) [5], the proposed two-stage algorithm (`Proposed`), and the `Proposed` algorithm refined by the `Ono` algorithm (`Proposed` + `refinement`). As discussed in Section VI, the `Proposed` algorithm may not recover the locations if the timing information is not estimated accurately. However, the estimated locations are usually closer to the true values than random initializations. Using them as an initial guess for the `Ono` algorithm, which estimates all the parameters simultaneously, may lead to a better estimation. The parameters used in the `Proposed` algorithm are listed in Table II. These parameters will be used throughout the experiment unless otherwise stated. In the `Ono` algorithm, the maximum iteration number is set to 2×10^5.

B. Convergence Analysis

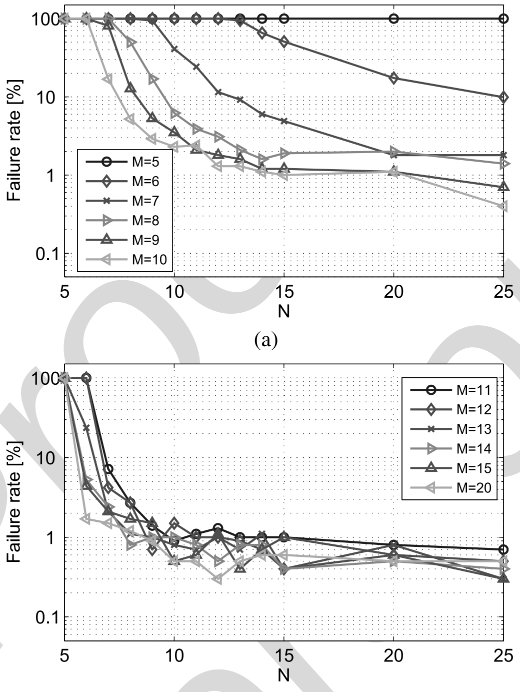
To investigate the convergence behaviour of the timing information estimation (Gauss-Newton low-rank approximation) algorithm, we test all the possible combinations of M and N. For each (M, N) configuration, we implement 1000 realizations. To reach global convergence, we try at most 10000 initializations in each realization. As described in Algorithm 1, a global convergence is detected when $\zeta_o < \zeta_{T2}$. If for one realization the algorithm can reach global convergence within 10000 initializations, we regard this realization as success and record the corresponding number of initializations as I_G. Otherwise, we regard it as failure. The failure rate is defined as

$$R_F(M, N) = \frac{N_F(M, N)}{1000} \times 100\%, \tag{41}$$

where $N_F(M, N)$ denotes the number of failed realizations for a specific (M, N) configuration.

TABLE II
Parameters Used in the Proposed Algorithm.

Parameter	Value	Parameter	Value
m_{T1}	1000	m_{T3}	50
m_{T2}	100	m_{T4}	100
ζ_{T1}	10^5	ν_{T1}	10^5
ζ_{T2}	10^{-15}	ν_{T1}	10^{-15}
$d_{\rho T}$	10^{-9}	$d_{\beta T}$	10^{-9}
λ	10^9	$d_{\max}$	10 m



(a)

(b)

Fig. 4. Failure rate for the timing information estimation when varying the number of sensors (M) and sources (N). (a) $M \in [5, 10]$. (b) $M \in [11, 20]$.

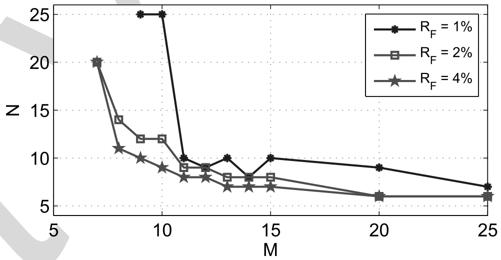


Fig. 5. Required minimum number of sensors (M) and sources (N) by the timing information estimation algorithm to achieve a failure rate of 1%, 2%, and 4%.

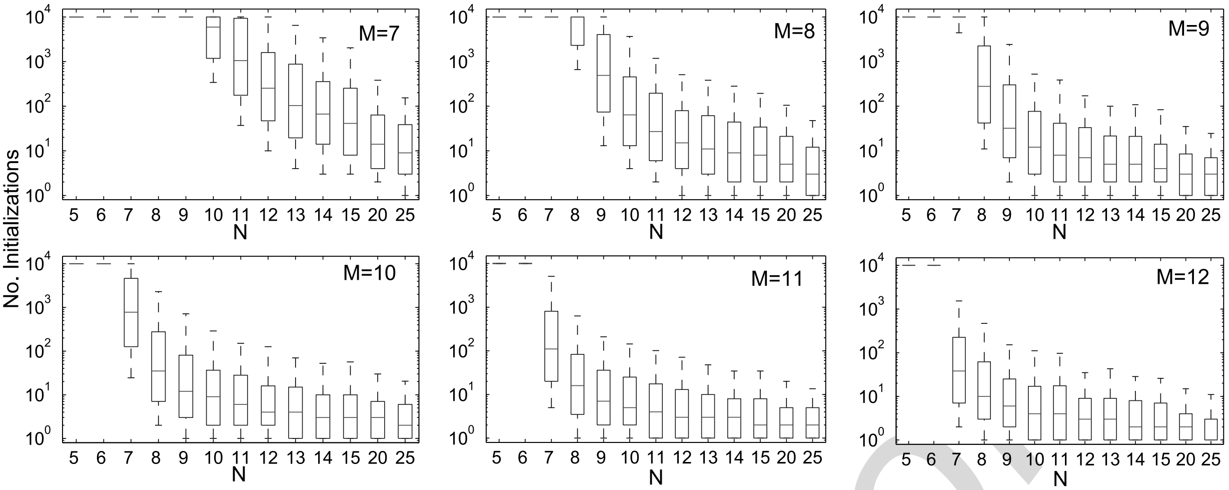


Fig. 6. Number of initializations (I_G) tried by the timing information estimation algorithm before reaching global convergence.

Each curve in Fig. 4 depicts the failure rate for a fixed M and variable N. Overall, the algorithm shows a high failure rate for small M and N, whereas the failure rate drops significantly when M and N increase. For the non-convex cost function (16) an over-constrained case with more observations than unknown parameters might reduce the number of local minima and lead to better convergence. The algorithm fails when $M = 5$, regardless of the value of N. When $M = 6$, the algorithm reaches global convergence even if the failure rate is still high. When $M \geq 7$, it becomes easier to reach global convergence. A fairly low failure rate (e.g., $< 2\%$) can be achieved if N is large enough. Moreover, the minimum N that is required to reach a same failure rate decreases when increasing M (cf. Fig. 5). When $M \geq 11$, the failure-rate curves become similar to each other. A low failure rate can be easily reached. In addition to this, when M and N are both large, the failure rate is slightly higher than 0, e.g., $R_F = 0.5\%$ for $M = 20$, $N = 25$.

Fig. 5 gives the minimum N that is required for reaching a defined failure rate ($R_F < 1\%$, 2%, 4%) for different M. It is observed that the lower the failure rate, the larger M and N are required. For the same failure rate the required minimum N decreases with M. This can be observed when $M > 10$.

Fig. 6 provides box-plots of the number of initializations (I_G) tried by the algorithm before reaching global convergence in 1000 realizations. The bottom and top of the box denote the 0.1 and 0.9 quantile, respectively. The line in the middle of the box denotes the median value. The bars represent the extreme values excluding the outliers, which are not shown in the figure.

In global, I_G decreases when M and N increase. This can be clearly observed from the variation of the median value of I_G in each panel, where M is fixed and N is varying. It can be further observed in each panel that, for a fixed M, the median value of I_G equals 10000 when N is small (the algorithm diverges) and then decreases significantly with increasing N. The decrease slows when N is large enough.

The span of the box (including 10% – 90% of the data) indicates that the performance of the algorithm may vary with different realizations of an (M, N) configuration. The height of the box is also related to M and N, and typically decreases when M and N increase. For instance, the box for $M = N = 8$ spans an interval [2000, 10000] while the box for $M = N = 12$ spans an interval [1, 10]. This indicates that it is easier for the algorithm to converge globally when M and N are large enough.

Fig. 6 can provide a reference for determining the maximum number of initializations (m_{T1}), a parameter used in Algorithm 1. For instance, we choose $m_{T1} = 1000$ in the remaining experiment since in most (M, N) configurations the algorithm can converge globally within 1000 initializations.

C. Performance Comparison

The testing data is generated using an equal number of sensors and sources ($M = N$) varying from 6 to 15. For each (M, N), we implement 200 realizations. Three algorithms are considered: $\texttt{Ono}$, $\texttt{Proposed}$, $\texttt{Proposed} + \texttt{refinement}$. Given the ground truth $\boldsymbol{R}$, $\boldsymbol{S}$ and the estimates $\tilde{\boldsymbol{R}}$, $\tilde{\boldsymbol{S}}$, the estimation error is calculated as

$$E_{\mathrm{rs}} = \frac{\sum_{i=1}^{M} \|\boldsymbol{r}_i - \tilde{\boldsymbol{r}}_i\| + \sum_{j=1}^{N} \|\boldsymbol{s}_j - \tilde{\boldsymbol{s}}_j\|}{M + N}. \qquad (42)$$

We assume that localization accuracy of 0.001 m is enough in practical applications, and thus we lower bound E_{rs} as $E_{\mathrm{rs}} = \max(0.001, E_{\mathrm{rs}})$. In order to account for the intrinsic translational and rotational freedom of the solution, the estimated and true sensor and source positions are optimally aligned by means of Procrustes Analysis [34], which finds the optimal translation and rotation of the solution that minimizes the sum of the distances between each pair of estimated and true positions.

The results are presented in Fig. 7, using the same box-plotting scheme as in Fig. 6. For readability, the estimation error is upper bounded at 3 m in Fig. 7. The overall performance can be ranked as $\texttt{Ono} < \texttt{Proposed} < \texttt{Proposed} + \texttt{refinement}$.

The median value of the estimation errors of $\texttt{Ono}$ does not vary significantly with M and N. However, the span of the box increases with increasing M and N. When M and N are large (e.g., > 10), the bottom of the box may possibly reach the lowest estimation bound, i.e., 0.001 m.

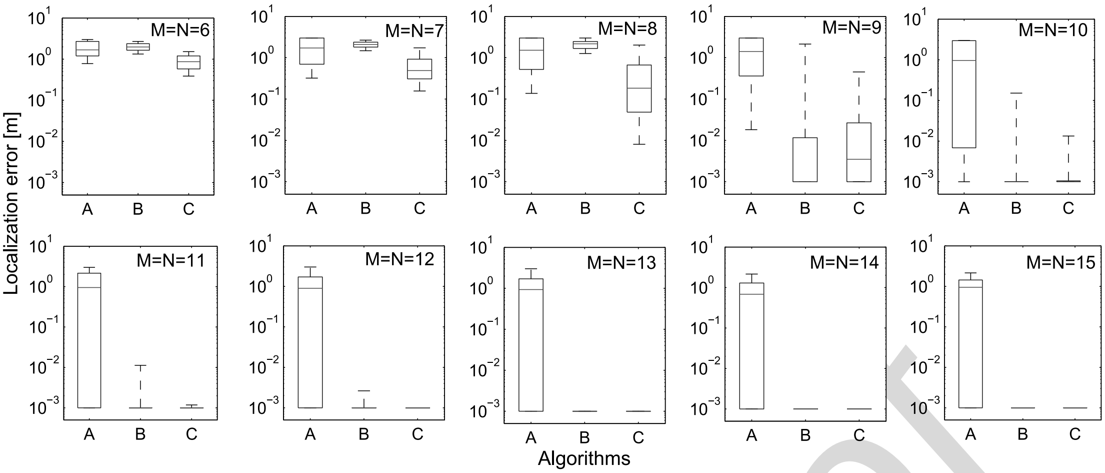


Fig. 7. Localization errors (E_{rs}) for different configurations of M and N: A: `Ono`; B: `Proposed`; C: `Proposed` + `refinement`.

`Proposed` performs slightly worse than `Ono` for $M = N = 8$, where both algorithms achieve large errors. The performance of `Proposed` improves evidently with increasing M and N. When $M, N \geq 10$, the median estimation error can reach the lowest estimation bound. The span of the box is still wide when $M = N = 10$, but decreases with increasing M and N. When $M, N \geq 13$, the span of the box is close to 0.

`Proposed` + `refinement` performs best among all the algorithms. For $M = N = 8$, it outperforms the other two algorithms although all three algorithms achieve large estimation errors. For $M, N \geq 12$, `Proposed` + `refinement` achieves a very low estimation error in all cases. For $M, N \geq 13$, `Proposed` performs similarly to `Proposed` + `refinement`.

In summary, `Proposed` significantly outperforms `Ono`. `Proposed` + `refinement` outperforms `Proposed` especially when M and N are small.

D. Robustness Analysis

The testing data is generated using equal number of sensors and sources ($M = N$) chosen from [8, 10, 12, 15]. For each (M, N) configuration, we implement 200 realizations. In each realization, the TDOA measurements are corrupted by Gaussian noise with zero mean and a standard deviation $\sigma \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ s. Apart from the considered algorithms, the Cramér-Rao bound (CRB) is calculated as a reference, using the method presented in [13]. The CRB provides a theoretical lower bound on the variance of any estimator, irrespective of the particular estimation method used. We average the estimation errors across 200 realizations:

$$E_{\mathrm{avg}} = 10^{\frac{1}{N_r}\sum_{r=1}^{N_r} \log_{10} E_{\mathrm{rs}}(r)}, \tag{43}$$

where $E_{\mathrm{rs}}(r)$ denotes the estimation error of the r-th realization, $N_r = 200$ is the number of realizations, and the log operation is introduced to reduce the influence of the highly dynamic range of the estimation error. Fig. 8 shows E_{avg} for different TDOA measurement errors (for readability E_{avg} is upper bounded to 3 m). Overall, the performance of the algorithms can still be ranked as `Ono` < `Proposed` < `Proposed` + `refinement`.

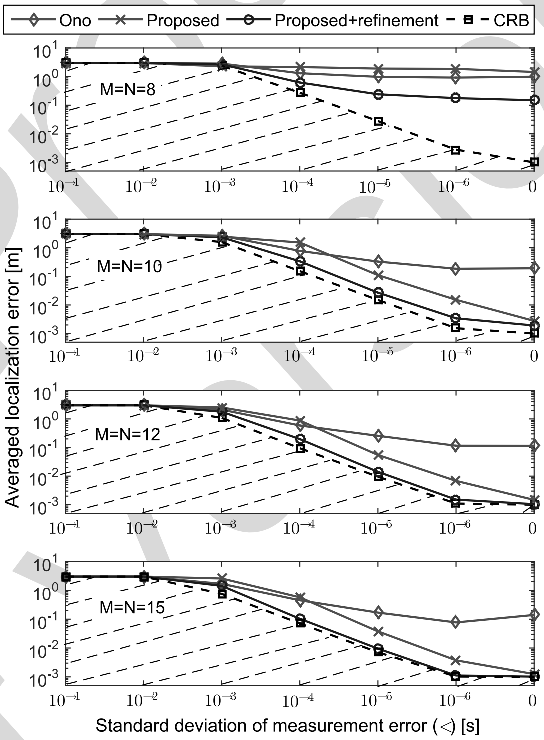


Fig. 8. Average localization errors (E_{avg}) by the considered algorithms in case of TDOA measurement errors. CRB represents the theoretical lower bound of the estimation.

CRB is influenced by M and N only slightly. It mainly depends on the standard deviation of the measurement error. The E_{avg} of CRB decreases when σ decreases. When $\sigma > 10^{-3}$ s, all the algorithms fail. When 10^{-4} s $\leq \sigma \leq 10^{-6}$ s, the performance of all the algorithms improves when σ decreases. `Ono` generally performs worst among all the algorithms. `Proposed` performs slightly worse than `Ono` when $M = N = 8$ but outperforms `Ono` in other cases. `Proposed` + `refinement` performs best. Its performance curve is far from the CRB curve when $M = N = 8$, close to the CRB curve when $M, N \geq 10$, and almost overlapped with the CRB curve when $M = N = 15$. Therefore, `Proposed` + `refinement` outperforms `Proposed` significantly in case of TDOA errors.

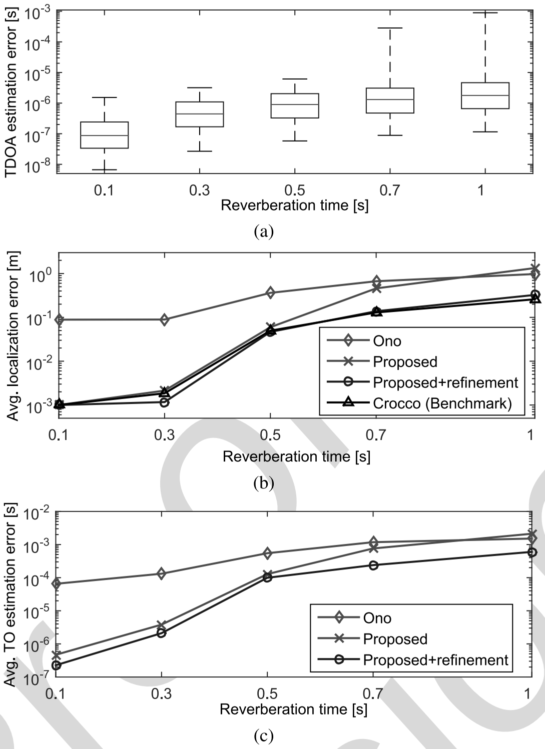
E. Realistic Simulation

We evaluate the performance of the proposed algorithm with realistic TDOA data estimated from simulated ambient sounds. Similarly to previous experiments, we consider a room of size $10 \text{ m} \times 10 \text{ m} \times 3 \text{ m}$, where sensors and sources are randomly distributed with $d_{\max} = 10$ m. The number of sensors and sources are both 15. The impulse responses from the sources to the sensors are simulated by the image-source method [38] with the reverberation time, RT_{60}, controlled by varying the absorption coefficients. The sound speed is 342 m/s and the sampling rate 48 kHz. The sound sources consist of nonoverlapping uttered speech signals, each occupying a time slot of 5 s with the length of speech 3 s. The onset time of each sound source is randomly chosen within the first two seconds of its time slot. The microphone signals are generated by convolving clean speech signals with room impulse responses. The capture time of each microphone is chosen randomly and uniformly from the interval $[-1, 1]$ s. We simulate 5 different reverberant scenarios with reverberation times 0.1 s, 0.3 s, 0.5 s, 0.7 s and 1 s, respectively. For each reverberation time we implement 10 realizations.

The acoustic scenario we consider is a simple case which assumes no overlaps between sound sources and the segmentation of each sound source in the microphone signals can be easily identified. It should be noted that, in practical applications, to find the data association between each sound source and the microphone signals is a nontrivial task, especially when multiple sources are simultaneously active. Although multiple target tracking methods [40], [41] have shown potential to address the data association problem, it still remains an open problem in real-world scenarios.

In this experiment the TDOA of each source with respect to a pair of microphones is estimated from the corresponding segmentations in the microphone signals. A coarse-to-fine scheme [5] is employed to estimate the TDOA, where the two microphone signals are coarsely aligned at first and then processed with the generalized cross-correlation with phase transform (GCC-PHAT) algorithm [39], which is well-known for its robustness to room reverberation. In the GCC-PHAT algorithm, we use a frame length of 8192 with half overlap. As shown in the robustness analysis in Section VII-D, `Proposed` works well when the TDOA estimation error is below 10^{-6} s. In order to achieve such an estimation accuracy, we search in the TDOA space with a small step size of 10^{-9} s.

We use `Ono`, `Proposed` and `Proposed` + `refinement` to estimate the sensor and source locations. For reference, we use the localization results obtained by the Crocco algorithm [13] as `Benchmark`. The Crocco algorithm works on TOA and assumes that both source onset times and device capture times are known. We use the general solution in the Crocco algorithm, which is essentially the same as the geometrical information estimation part of the `Proposed` algorithm. To apply the Crocco algorithm, we derive the TOA from the TDOA (which is estimated from the microphone signals) and the distance between the first microphone and all sources (cf. (1) and (2)), and assume the time offsets between devices to be known. In this way, the estimation errors contained in the estimated TOAs and TDOAs are comparable.



(a)

(b)

(c)

Fig. 9. Performance evaluation for the considered algorithms using TDOAs estimated from ambient sounds in different reverberant scenarios. The numbers of sensors and sources are both 15. (a) TDOA estimation. (b) Sensor and source localization. (c) Time offset (TO) estimation.

We calculate the TDOA estimation errors for all the sensor and source combinations that are contained in the TDOA matrix. The TDOA estimation error is defined as the absolute difference between the estimated value and the ground truth, which is obtained from the prior knowledge of the sensor and source locations and the device capture times. For each reverberation time, we collect all the TDOA estimation errors in 10 realizations and present them in Fig. 9(a), using the same box-plotting scheme as in Fig. 6. It is clearly observed in Fig. 9(a) that the median value of the TDOA estimation errors increases with reverberation time (RT). More outliers (i.e., larger errors) are observed when $\mathrm{RT} \geq 0.7$ s. For instance, the median value of the errors at $\mathrm{RT} = 1$ s is around 10^{-6} s, whereas the upper bound of the error can reach 10^{-3} s.

The average localization errors (E_{avg} in (43)) obtained by the algorithms in different reverberant scenarios are depicted in Fig. 9(b). Overall, the performance degrades with reverberation time, and can be ranked as `Ono` < `Proposed` < `Proposed` + `refinement` ≈ `Benchmark`. The Crocco (`Benchmark`) algorithm, which assumes timing information to be known, also shows localization errors due to large TDOA estimation errors in high reverberation. As shown in Fig. 9(b), `Proposed` can estimate the locations very accurately when $\mathrm{RT} \leq 0.3$ s. However, its E_{avg} rises to 0.05 m at $\mathrm{RT} = 0.5$ s, and becomes larger than 0.5 m when $\mathrm{RT} \geq 0.7$ s. `Proposed` + `refinement` can significantly improve the localization performance in high reverberation, with E_{avg} being around 0.1 m and 0.2 m for $\mathrm{RT} = 0.7$ s and

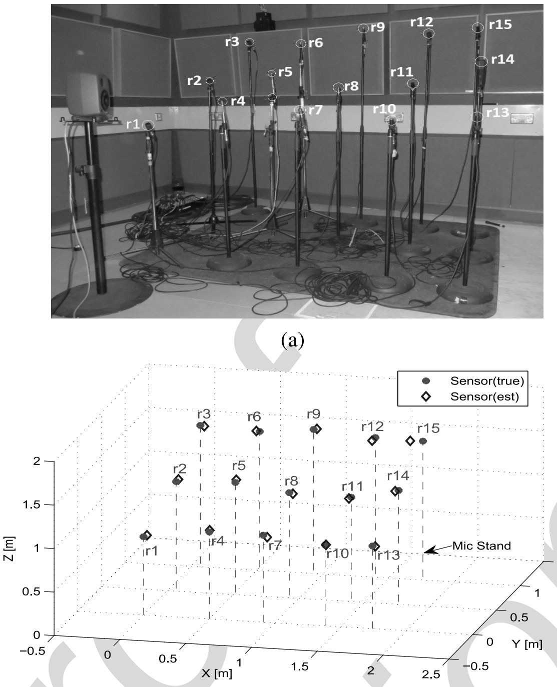
$\mathrm{RT} = 1$ s, respectively. $\texttt{Proposed} + \texttt{refinement}$ can achieve similar performance as $\texttt{Benchmark}$.

With the estimated sensor and source locations, the time offsets between the first microphone and other microphones, as by-products, can be easily calculated using (2). We define the offset time estimation error as the absolute difference between the estimated value and the ground truth, and calculate the average offset time estimation error for 10 realizations, using an equation similar to (43). The results are shown in Fig. 9(c). Since the time offsets are calculated from the sensor and source locations, the performance of time offset estimation depends significantly on sensor and source localization. As shown in Fig. 9(c), the time offset estimation error curves of all the algorithms show similar variation trends to the localization error curves in Fig. 9(b). $\texttt{Proposed} + \texttt{refinement}$ can achieve time offset estimation errors below one sample (around 20 μs at sampling rate 48 kHz) when $\mathrm{RT} \le 0.3$ s. The estimation error rises to around 5 samples (0.1 ms) and 25 samples (0.5 ms) at $\mathrm{RT} = 0.5$ s and $\mathrm{RT} = 1$ s, respectively.

F. Real Recording

We present an example of using the considered algorithms with real-recorded data in a listening room of size 6 m $\times$ 5 m $\times$ 3 m and with a reverberation time of around 200 ms. We use 15 microphones of various types such as Sennheiser E600, DPA 4006-TL and Beyerdynamic MCE530, placed as shown in Fig. 10(a). The microphones are connected to an RME Fireface 800 through two Focusrite ISA828 microphone preamplifiers (8 channels each). The sampling rate is 44.1 kHz. The same testing speech used in Section VII-E is played by a Genelec 8010 loudspeaker at 18 different locations (acting as 18 sound sources). To simulate unknown device capture times, a delay randomly chosen from the interval $[-1, 1]$ s is applied to the recording from each microphone. These recordings are used in the localization procedure. We only have the ground truth of the microphone locations, which are manually measured. We investigate microphone localization performance with this ground truth. Although the microphones are placed with a relatively regular topology, in practice the considered algorithms can be used for arbitrary microphone placement.

Compared with the simulated (image-source) scenarios, the performance of TDOA estimation with real recordings may be degraded by extra reflections from microphone stands and also by the fact that the loudspeaker is not a point source. For localization we applied the $\texttt{Ono}$, $\texttt{Proposed}$ and $\texttt{Proposed} + \texttt{refinement}$ algorithms. The average localization errors obtained by the three algorithms are about 0.39 m, 0.03 m and 0.03 m, respectively. The true and estimated microphone locations ($\texttt{Proposed}$) are shown in Fig. 10(b). It is clearly observed that the topology of the microphone network can be recovered. The consistence between the true and estimated microphone locations confirms the potential of the proposed algorithms in real-world applications.



(a)

(b)

Fig. 10. Microphone localization with real-recorded data. (a) Recording environment. (b) Localization result by the $\texttt{Proposed}$ algorithm. The average localization error of all the microphones is about 0.03 m.

VIII. Conclusions

We proposed a general solution for sensor and source localization in an ad-hoc array using TDOAs which are biased by unknown time offsets. To overcome the local minima problem when estimating the unknown timing information, we derived the boundary of the timing parameters and utilized a multi-initialization scheme. These strategies can minimize the influence of local minima efficiently. Simulation results demonstrate that the $\texttt{Proposed}$ algorithm outperforms existing algorithms in most cases. The performance of the $\texttt{Proposed}$ algorithm can be further improved by a refinement strategy. Error analysis demonstrates that the $\texttt{Proposed} + \texttt{refinement}$ algorithm can get close to the Cramér-Rao bound with at least 10 sensors and sources. Experiments using TDOAs estimated from simulated and real-recorded speech data confirm the potential of the $\texttt{Proposed}$ algorithm in real-world applications. Specifically, the $\texttt{Proposed} + \texttt{refinement}$ algorithm can achieve high-quality estimation for speech data simulated at reverberation time 0.3 s and the estimation error rises to 0.05 m, 0.1 m and 0.2 m for reverberation times 0.5 s, 0.7 s and 1 s, respectively. For real-recorded data in an environment with reverberation time of 0.2 s, the $\texttt{Proposed} + \texttt{refinement}$ algorithm can achieve an estimation error of 0.03 m.

The $\texttt{Proposed}$ algorithm however requires a minimum number of sensors and sources (e.g., $M, N \ge 10$) to guarantee its performance. Although the performance improves significantly as M and N increase, the local minima problem is still not completely solved, e.g., it can still be observed with a low probability even if M and N are large. An interesting direction for future research would be to incorporate prior knowledge of the ad-hoc array into the $\texttt{Proposed}$ algorithm, so that the dependence on the number of sensors and sources can be relieved and the local minima problem can be fully overcome. Moreover, although the $\texttt{Proposed}$ $(+\texttt{refinement})$ algorithm shows promising results in reverberant scenarios, the robustness to TDOA estimation outliers could be improved by exploiting the redundancy of the TDOA information when using a large number of sensors and sources.

Appendix A
Jacobian Matrix in (19)

For $\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\zeta}_A \\ \lambda\boldsymbol{\zeta}_B \end{bmatrix}$ and $\boldsymbol{\rho} = \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\eta} \\ \boldsymbol{x} \end{bmatrix}$, the Jacobian matrix $\boldsymbol{J} = \frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\rho}}$, which is of size $Q \times (M + N - 1 + 3(N - 4))$, can be calculated as

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial \boldsymbol{\zeta}_A}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{\zeta}_A}{\partial \boldsymbol{\eta}} & \frac{\partial \boldsymbol{\zeta}_A}{\partial \boldsymbol{x}} \\ \frac{\lambda\partial \boldsymbol{\zeta}_B}{\partial \boldsymbol{\delta}} & \frac{\lambda\partial \boldsymbol{\zeta}_B}{\partial \boldsymbol{\eta}} & \frac{\lambda\partial \boldsymbol{\zeta}_B}{\partial \boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{J}_{12} & \boldsymbol{J}_{13} \\ \lambda\boldsymbol{J}_{21} & \lambda\boldsymbol{J}_{22} & \lambda\boldsymbol{J}_{23} \end{bmatrix}.$$

The block matrices can be computed as described below.

$$\boldsymbol{J}_{11} = \begin{bmatrix} \frac{\partial[\boldsymbol{U}]_{11}}{\partial\delta_1} & \cdots & \frac{\partial[\boldsymbol{U}]_{11}}{\partial\delta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial\delta_1} & \cdots & \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial\delta_M} \end{bmatrix}.$$

Based on the definition of $\boldsymbol{U}$ in (7), the element-wise differential is expressed as

$$\frac{\partial[\boldsymbol{U}]_{i-1,j-1}}{\partial\delta_k} = \begin{cases} (2\eta_j - 2(t_{1j} - t_{11}))\cdot \uparrow_{i,k}, & k = 1 \\ (2(t_{ij} - t_{i1}) - 2\eta_j)\cdot \uparrow_{i,k}, & k = 2, \ldots, M \end{cases} \tag{A1}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$. Here $\uparrow_{i,k}= \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$.

$$\boldsymbol{J}_{12} = \begin{bmatrix} \frac{\partial[\boldsymbol{U}]_{11}}{\partial\eta_1} & \cdots & \frac{\partial[\boldsymbol{U}]_{11}}{\partial\eta_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial\eta_1} & \cdots & \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial\eta_N} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{U}]_{i-1,j-1}}{\partial\eta_k} = -(2(t_{ij} - t_{1j}) + 2\delta_i)\cdot \uparrow_{j,k}, \tag{A2}$$

for $i = 2, \ldots, M$, $j = 2, \ldots, N$, and $k = 2, \ldots, N$.

$$\boldsymbol{J}_{13} = \begin{bmatrix} \frac{\partial[\boldsymbol{U}]_{11}}{\partial[\boldsymbol{X}]_{11}} & \cdots & \frac{\partial[\boldsymbol{U}]_{11}}{\partial[\boldsymbol{X}]_{3,N-4}} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial[\boldsymbol{X}]_{11}} & \cdots & \frac{\partial[\boldsymbol{U}]_{(M-1),(N-1)}}{\partial[\boldsymbol{X}]_{3,N-4}} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{U}]_{ij}}{\partial[\boldsymbol{X}]_{kl}} = 0, \tag{A3}$$

for $i = 1, \ldots, M - 1$, $j = 1, \ldots, N - 1$, $k = 1, \ldots, 3$, and $l = 1, \ldots, N - 4$.

Let $\boldsymbol{V} = (\boldsymbol{A} + \boldsymbol{F})\boldsymbol{X} - \boldsymbol{B} - \boldsymbol{G}$.

$$\boldsymbol{J}_{21} = \begin{bmatrix} \frac{\partial[\boldsymbol{V}]_{11}}{\partial\delta_1} & \cdots & \frac{\partial[\boldsymbol{V}]_{11}}{\partial\delta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial\delta_1} & \cdots & \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial\delta_M} \end{bmatrix},$$

where $\frac{\partial[\boldsymbol{V}]_{ij}}{\partial\delta_k} = \frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{ij}}{\partial\delta_k}$.

Since $[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{ij} = \sum_{u=1}^{3}[\boldsymbol{U}]_{iu}[\boldsymbol{X}]_{uj} - [\boldsymbol{U}]_{i,j+3}$, it follows

$$\frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{i-1,j-1}}{\partial\delta_1} = \sum_{u=1}^{3}(2\eta_u - 2(t_{1u} - t_{11}))[\boldsymbol{X}]_{uj} - \{(2\eta_{j+3} - 2(t_{1,j+3} - t_{11})\}, \tag{A4}$$

and

$$\frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{i-1,j-1}}{\partial\delta_k} = \uparrow_{i,k}\cdot \left\{\left(\sum_{u=1}^{3}(2(t_{iu} - t_{i1}) - 2\eta_u)[\boldsymbol{X}]_{uj}\right) + 2\eta_{j+3} - 2(t_{i,j+3} - t_{i1})\right\}, \tag{A5}$$

for $i = 2, \ldots, M$, $j = 2, \ldots, N$, and $k = 2, \ldots, M$.

$$\boldsymbol{J}_{22} = \begin{bmatrix} \frac{\partial[\boldsymbol{V}]_{11}}{\partial\eta_1} & \cdots & \frac{\partial[\boldsymbol{V}]_{11}}{\partial\eta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial\eta_1} & \cdots & \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial\eta_M} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{V}]_{ij}}{\partial\eta_k} = \frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{ij}}{\partial\eta_k}.$$

$$\frac{\partial[\boldsymbol{F}\boldsymbol{X} - \boldsymbol{G}]_{i-1,j-1}}{\partial\eta_k} = \begin{cases} (-2(t_{ij} - t_{1j}) + 2(\delta_1 - \delta_i))[\boldsymbol{X}]_{kj}, & k=2, \ldots, 4 \\ (-2(t_{i,j+3} - t_{1,j+3}) + 2(\delta_1 - \delta_i))\cdot \uparrow_{k,j+3}, & k=5, \ldots, N \end{cases} \tag{A6}$$

for $i = 2, \ldots, M$ and $j = 2, \ldots, N$.

$$\boldsymbol{J}_{23} = \begin{bmatrix} \frac{\partial[\boldsymbol{V}]_{11}}{\partial[\boldsymbol{X}]_{11}} & \cdots & \frac{\partial[\boldsymbol{V}]_{11}}{\partial[\boldsymbol{X}]_{3,N-4}} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial[\boldsymbol{X}]_{11}} & \cdots & \frac{\partial[\boldsymbol{V}]_{(M-1),(N-4)}}{\partial[\boldsymbol{X}]_{3,N-4}} \end{bmatrix},$$

where $\frac{\partial[\boldsymbol{V}]_{ij}}{\partial[\boldsymbol{X}]_{kl}} = \frac{\partial[(\boldsymbol{A} + \boldsymbol{F})\boldsymbol{X}]_{ij}}{\partial[\boldsymbol{X}]_{kl}}$.

Since $[(\boldsymbol{A} + \boldsymbol{F})\boldsymbol{X}]_{ij} = \sum_{u=1}^{3}(\boldsymbol{A} + \boldsymbol{F})_{iu}[\boldsymbol{X}]_{uj}$,

$$\frac{\partial[(\boldsymbol{A} + \boldsymbol{F})\boldsymbol{X}]_{ij}}{\partial[\boldsymbol{X}]_{kl}} = \uparrow_{l,j} [\boldsymbol{A} + \boldsymbol{F}]_{ik} \tag{A7}$$

for $i = 1, \ldots, M - 1$, $j = 1, \ldots, N - 1$, $k = 1, \ldots, 3$, and $l = 1, \ldots, N - 4$.

Appendix B
Jacobian Matrix in (39)

For $\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_A \\ \boldsymbol{\nu}_B \end{bmatrix}$ and $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{c} \\ s_{x1} \end{bmatrix}$, the Jacobian matrix $\boldsymbol{I} = \frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\beta}}$, which is of size $(M + N - 2) \times 10$, can be calculated as

$$\boldsymbol{I} = \begin{bmatrix} \frac{\partial \boldsymbol{\nu}_A}{\partial \boldsymbol{c}} & \frac{\partial \boldsymbol{\nu}_A}{\partial s_{x1}} \\ \frac{\partial \boldsymbol{\nu}_B}{\partial \boldsymbol{c}} & \frac{\partial \boldsymbol{\nu}_B}{\partial s_{x1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{11} & \boldsymbol{I}_{12} \\ \boldsymbol{I}_{21} & \boldsymbol{I}_{22} \end{bmatrix}.$$

The block matrices can be computed as described below.

Based on the definitions in (30) and (38), it follows

$$[\boldsymbol{\nu}_A]_{i-1} = \sum_{u=1}^{3} [\boldsymbol{D_L C}]_{iu}^2 - 2[\boldsymbol{D_L C}]_{i1} s_{x1} - \bar{g}_i \tag{B1}$$

for $i = 2, \ldots, M$.

$$\boldsymbol{I}_{11} = \begin{bmatrix} \frac{\partial[\boldsymbol{\nu}_A]_1}{\partial[\boldsymbol{C}]_{11}} & \cdots & \frac{\partial[\boldsymbol{\nu}_A]_1}{\partial[\boldsymbol{C}]_{33}} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{\nu}_A]_{M-1}}{\partial[\boldsymbol{C}]_{11}} & \cdots & \frac{\partial[\boldsymbol{\nu}_A]_{M-1}}{\partial[\boldsymbol{C}]_{33}} \end{bmatrix},$$

where

$$\frac{\partial[\boldsymbol{\nu}_A]_{i-1}}{\partial[\boldsymbol{C}]_{kl}} = 2[\boldsymbol{D}_L\boldsymbol{C}]_{il}[\boldsymbol{U}]_{ik} - 2\uparrow_{l,1} \boldsymbol{U}_{ik} s_{x1} \tag{B2}$$

for $i = 2, \ldots, M$, $k = 1, 2, 3$, and $l = 1, 2, 3$.

$$\boldsymbol{I}_{12} = \left[\frac{\partial[\boldsymbol{\nu}_A]_1}{\partial s_{x1}} \quad \cdots \quad \frac{\partial[\boldsymbol{\nu}_A]_{M-1}}{\partial s_{x1}} \right]^{\mathrm{T}},$$

where

$$\frac{\partial[\boldsymbol{\nu}_A]_i}{\partial s_{x1}} = -2[\boldsymbol{D_L C}]_{i1} \tag{B3}$$

for $i = 1, \ldots, M-1$.

Based on the definitions in (38), it follows

$$[\boldsymbol{\nu}_B]_{j-1} = \sum_{u=1}^{3} [\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}}]_{uj}^2 + 2[\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}}]_{1j} s_{x1} - \bar{\bar{g}}_j, \tag{B4}$$

for $j = 2, \ldots, N$.

$$\boldsymbol{I}_{21} = \begin{bmatrix} \frac{\partial[\boldsymbol{\nu}_B]_1}{\partial[\boldsymbol{C}]_{11}} & \cdots & \frac{\partial[\boldsymbol{\nu}_B]_1}{\partial[\boldsymbol{C}]_{33}} \\ \vdots & \ddots & \vdots \\ \frac{\partial[\boldsymbol{\nu}_B]_{N-1}}{\partial[\boldsymbol{C}]_{11}} & \cdots & \frac{\partial[\boldsymbol{\nu}_B]_{N-1}}{\partial[\boldsymbol{C}]_{33}} \end{bmatrix},$$

where

$$\begin{aligned} \frac{\partial[\boldsymbol{\nu}_B]_j}{[\boldsymbol{C}]_{kl}} &= 2[\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}}]_{\cdot,j}^{\mathrm{T}} \frac{\partial\boldsymbol{C}^{-1}}{\partial[\boldsymbol{C}]_{kl}} [\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}}]_{\cdot,j} \\ &\quad + 2s_{x1} \frac{\partial\boldsymbol{C}^{-1}}{\partial[\boldsymbol{C}]_{kl}} [\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}}]_{\cdot,j}, \end{aligned} \tag{B5}$$

for $j = 1, \ldots, N-1$, $k = 1, 2, 3$, and $l = 1, 2, 3$. Using the inversion theory, we have

$$\frac{\partial\boldsymbol{C}^{-1}}{\partial[\boldsymbol{C}]_{kl}} = -\boldsymbol{C}^{-1}\frac{\partial\boldsymbol{C}}{\partial[\boldsymbol{C}]_{kl}}\boldsymbol{C}^{-1} = -\boldsymbol{C}^{-1}\boldsymbol{O}_{kl}\boldsymbol{C}^{-1}, \tag{B6}$$

where $\boldsymbol{O}_{kl}$ is a matrix with the same size of $\boldsymbol{C}$, with the (k, l)-th element being 1 and other elements being 0.

$$\boldsymbol{I}_{22} = \left[\frac{\partial[\boldsymbol{\nu}_B]_1}{\partial s_{x1}} \quad \cdots \quad \frac{\partial[\boldsymbol{\nu}_B]_{N-1}}{\partial s_{x1}} \right]^{\mathrm{T}},$$

where, for $j = 1, \ldots, N-1$,

$$\frac{\partial[\boldsymbol{\nu}_B]_j}{\partial s_{x1}} = 2[\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_3\boldsymbol{D}_R^{\mathrm{T}}]_{j1}. \tag{B7}$$

REFERENCES

[1] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *Proc. IEEE Symp. Communications and Vehicular Technol. in the Benelux*, Ghent, Belgium, 2011, pp. 1–6.

[2] M. H. Hennecke and G. A. Fink, "Towards acoustic self-localization of ad hoc smartphone arrays," in *Proc. IEEE Joint Workshop on Hands-Free Speech Communication and Microphone Arrays*, Edinburgh, UK, 2011, pp. 127–132.

[3] L. Wang, "Multi-band multi-centroid clustering based permutation alignment for frequency-domain blind speech separation," *Digital Signal Process.*, vol. 31, pp. 79–92, Aug. 2014.

[4] L. Wang, T. Gerkmann, and S. Doclo, "Noise power spectral density estimation using MaxNSR blocking matrix," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 9, pp. 493–1508, Sep. 2015.

[5] N. Ono, H. Kohno, N. Ito, and S. Sagayama, "Blind alignment of asynchronously recorded signals for distributed microphone array," in *Proc. IEEE Workshop on Applicat. of Signal Process. to Audio and Acoustics*, New York, USA, 2009, pp. 161–164.

[6] S. T. Birchfield and A. Subramanya, "Microphone array position calibration by basis-point classical multidimensional scaling," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1025–1034, Sep. 2005.

[7] S. Wilson, J. Walters, and J. Abel, "Speaker locations from inter-speaker range measurements: Closed-form estimator and performance relative to the Cramer-Rao lower bound," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Montreal, Canada, 2004, pp. 389–392.

[8] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. Int. Conf. Embedded Networked Sensor Systems*, Baltimore, USA, 2004, pp. 50–61.

[9] M. J. Taghizadeh, R. Parhizkar, P. N. Garner, H. Bourlard, and A. Asaei, "Ad hoc microphone array calibration: Euclidean distance matrix completion algorithm and theoretical guarantees," *Signal Process.*, vol. 107, pp. 123–140, Feb. 2015.

[10] K. K. Chintalapudi, A. Dhariwal, R. Govindan, and G. Sukhatme, "Ad-hoc localization using ranging and sectoring," in *Proc. IEEE Infocom*, Hong Kong, China, 2004, pp. 2662–2672.

[11] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: A high accuracy acoustic ranging system using COTS mobile devices," in *Proc. Int. Conf. Embedded Networked Sensor Systems*, Sydney, Australia, 2007, pp. 1–14.

[12] J. Herrera and H. S. Kim, "Ping-pong: Using smartphones to measure distances and relative positions," in *Proc. Meetings on Acoustics*, San Francisco, USA, 2013, pp. 1–10.

[13] M. Crocco, A. Del Bue, and V. Murino, "A bilinear approach to the position self-calibration of multiple sensors," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 660–673, Feb. 2012.

[14] V. C. Raykar, I. V. Kozintsev, and R. Lienhart, "Position calibration of microphones and loudspeakers in distributed computing platforms," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 70–83, Jan. 2005.

[15] S. Thrun, "Affine structure from sound," in *Advances in Neural Information Processing System*, Y. Weiss, B. Scholkopf, and J. Platt, Eds. Cambridge, USA: MIT Press, 2005, pp. 1353–1360.

[16] R. Heusdens and N. D. Gaubitch, "Time-delay estimation for TOA-based localization of multiple sensors," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Florence, Italy, 2014, pp. 609–613.

[17] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, "Auto-localization in ad-hoc microphone arrays," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Vancouver, Canada, 2013, pp. 106–110.

[18] F. Jiang, Y. Kuang, and K. Astrom, "Time delay estimation for TDOA self-calibration using truncated nuclear norm regularization," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Vancouver, Canada, 2013, pp. 3885–3889.

[19] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, "Calibration of distributed sound acquisition systems using TOA measurements from a moving acoustic source," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Florence, Italy, 2014, pp. 7455–7459.

[20] I. McCowan, M. Lincoln, and I. Himawan, "Microphone array shape calibration in diffuse noise fields," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 3, pp. 666–670, Mar. 2008.

[21] M. J. Taghizadeh, P. N. Garner, and H. Bourlard, "Enhanced diffuse field model for ad hoc microphone array calibration," *Signal Process.*, vol. 101, pp. 242–255, Aug. 2014.

[22] P. Pertila, M. S. Hamalainen, and M. Mieskolainen, "Passive temporal offset estimation of multichannel recordings of an ad-hoc microphone array," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 11, pp. 2393–2402, Nov. 2013.

[23] V. Deleskog, H. Habberstad, G. Hendeby, D. Lindgren, and N. Wahlstrom, "Robust NLS sensor localization using MDS initialization," in *Proc. Int. Conf. Information Fusion*, Salamanca, Spain, 2014, pp. 1–7.

[24] M. Parviainen, P. Pertila, and M. S. Hamalainen, "Self-localization of wireless acoustic sensors in meeting rooms," in *Proc. Joint Workshop Hands-Free Speech Communication and Microphone Arrays*, Nancy, France, 2014, pp. 152–156.

[25] T. K. Hon, L. Wang, J. D. Reiss, and A. Cavallaro, "Audio fingerprinting for multi-device self-localization," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 10, pp. 1623–1636, Oct. 2015.

[26] R. Biswas and S. Thrun, "A passive approach to sensor network localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 1544–1549.

[27] M. Chen, Z. Liu, L. W. He, P. Chou, and Z. Zhang, "Energy-based position estimation of microphones and speakers for ad hoc microphone arrays," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, USA, 2007, pp. 22–25.

[28] Z. Liu, Z. Zhang, L. W. He, and P. Chou, "Energy-based sound source localization and gain normalization for ad hoc microphone arrays," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Honolulu, USA, 2007, pp. 761–764.

[29] X. Sheng and Y. H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, Jan. 2005.

[30] M. Pollefeys and D. Nister, "Direct computation of sound and microphone locations from time-difference-of-arrival data," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Las Vegas, USA, 2008, pp. 2445–2448.

[31] Y. Kuang, S. Burgess, A. Torstensson, and K. Astrom, "A complete characterization and solution to the microphone position self-calibration problem," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Vancouver, Canada, 2013, pp. 3875–3879.

[32] Y. Kuang and K. Astrom, "Stratified sensor network self-calibration from TDOA measurements," in *Proc. European Signal Process. Conf.*, Marrakech, Morroco, 2013, pp. 1–5.

[33] S. Zhayida, F. Andersson, Y. Kuang, and K. Astrom, "An automatic system for microphone self-localization using ambient sound," in *Proc. European Signal Process. Conf.*, Lisbon, Portugal, 2014, pp. 954–958.

[34] G. A. F. Seber, *Multivariate Observations*. Hoboken, USA: John Wiley & Sons Inc., 1984.

[35] J. Nocedal and S. Wright, *Numerical Optimization*. New York, USA: Springer, 1999.

[36] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. Annual ACM Symp. Theory of Computing*, Palo Alto, USA, 2013, pp. 665–674.

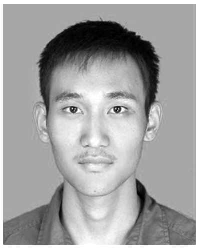
[37] S. Van Huffel, H. Park, and J. B. Rosen, "Formulation and solution of structured total least norm problems for parameter estimation," *IEEE Trans. Signal Process.*, vol. 44, no. 10, pp. 2464–2474, Oct. 1996.

[38] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, Apr. 1979.

[39] M. S. Brandstein and H. F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Munich, Germany, 1997, pp. 375–378.

[40] C. Hue, J. P. Le Cadre, and P. Perez, "Sequential Monte Carlo methods for multiple target tracking and data fusion," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 309–325, Feb. 2002.

[41] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian Multiple Target Tracking*. Boston, USA: Artech House, 2013.



Lin Wang received the B.S. degree in electronic engineering from Tianjin University, China, in 2003; and the Ph.D degree in signal processing from Dalian University of Technology, China, in 2010. From 2011 to 2013, he was an Alexander von Humboldt Fellow at the University of Oldenburg, Germany. Since 2014, he has been a postdoctoral researcher in the Centre for Intelligent Sensing at Queen Mary University of London. His research interests include video and audio compression, microphone array, blind source separation, and 3D audio processing.

Tsz-Kin Hon received his B.Eng. degree in electronic and computer engineering from the Hong Kong University of Science and Technology (HKUST) in 2006; and the Ph.D. degree in digital signal processing from Kings College London (KCL) in 2013. He was a Research Engineer in the R&D of the Giant Electronic Ltd. between 2006 and 2009. He is currently a Postdoctoral Research Assistant in the Centre for Intelligent Sensing at Queen Mary University of London. His research interests include audio and video signal processing, device localization and synchronization, multi-source signal processing, joint time–frequency analysis and filtering, acoustic echo cancellation, speech enhancement, and biomedical signal processing.

Joshua D. Reiss is a Reader in Audio Engineering with the Centre for Digital Music in the School of Electronic Engineering and Computer Science at Queen Mary University of London. He has bachelors degrees in both physics and mathematics, and earned his Ph.D. in physics from the Georgia Institute of Technology. He is a member of the Board of Governors of the Audio Engineering Society, and co-founder of the company MixGenius, now known as LandR. Dr. Reiss has published more than 100 scientific papers and serves on several steering and technical committees. He has investigated sound synthesis, time scaling and pitch shifting, source separation, polyphonic music transcription, loudspeaker design, automatic mixing for live sound, and digital audio effects. His primary focus of research, which ties together many of the above topics, is on the use of state-of-the-art signal processing techniques for professional sound engineering.

Andrea Cavallaro received the Ph.D. degree in electrical engineering from Swiss Federal Institute of Technology, Lausanne, Switzerland, in 2002. He was a Research Fellow with British Telecommunications in 2004. He is a Professor of Multimedia Signal Processing and the Director of the Centre for Intelligent Sensing at Queen Mary University of London. He has authored more than 150 journal and conference papers, one monograph on Video Tracking (Wiley, 2011), and three edited books, Multi-Camera Networks (Elsevier, 2009), Analysis, Retrieval and Delivery of Multimedia Content (Springer, 2012), and Intelligent Multimedia Surveillance (Springer, 2013). Prof. Cavallaro is an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING and a member of the editorial board of the IEEE MultiMedia Magazine. He is an elected member of the IEEE Image, Video, and Multidimensional Signal Processing Technical Committee, and is the Chair of its Awards Committee, and an elected member of the IEEE Circuits and Systems Society Visual Communications and Signal Processing Technical Committee. He served as an elected member of the IEEE Signal Processing Society Multimedia Signal Processing Technical Committee, as an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA and IEEE TRANSACTIONS ON SIGNAL PROCESSING, as an Associate Editor and as an Area Editor of IEEE Signal Processing Magazine, and as a Guest Editor of eleven special issues of international journals. He was General Chair for the IEEE/ACM ICDSC 2009, BMVC 2009, M2SFA2 2008, SSPE 2007, and IEEE AVSS 2007. He was Technical Program Chair of the IEEE AVSS 2011, the European Signal Processing Conference in 2008, and WIAMIS 2010. He received the Royal Academy of Engineering Teaching Prize in 2007, three Student Paper Awards on target tracking and perceptually sensitive coding at the IEEE ICASSP in 2005, 2007, and 2009, respectively, and the Best Paper Award at IEEE AVSS 2009.