Localizing Multiple Events Using Times of Arrival: a Parallelized, Hierarchical Approach to the Association Problem

Sriram Venkateswaran, *Member, IEEE*, and Upamanyu Madhow, *Fellow, IEEE*

***Abstract*—A fundamental problem in localizing multiple events based on Times of Arrival (ToAs) at a number of sensors is that of associating ToAs with events. We consider this problem in the context of acoustic sensors monitoring events that are closely spaced in time. Due to the relatively low speed of propagation of sound, the order in which the events arrive at a sensor need not be the same as the order in which they occur, potentially creating fundamental ambiguities. We first explore such ambiguities in an idealized setting with two events and noiseless observations, showing that it is possible to localize both events with nine or more sensors (as long as degenerate sensor placement is avoided), but that we can construct examples with six sensors for which unambiguous space-time localization is not possible. We then show that these potential ambiguities are not a bottleneck in typical practical settings, proposing and evaluating an algorithm that successfully localizes multiple events using noisy observations. The algorithm employs parallelism and hierarchical processing to avoid the excessive complexity of naïvely trying all possible associations of events with ToAs. We use discretization of hypothesized event times to enable us to efficiently generate a set of candidate event locations, which contain noisy versions of true events as well as *phantom* events. We refine these estimates iteratively, discarding "obvious" phantoms, and then solve a linear programming formulation for matching true events to ToAs, while identifying outliers and misses. Simulation results indicate excellent performance that is comparable to a genie-based algorithm which is given the correct association between ToAs and events.**

***Index Terms*—Multiple events, source localization, time of arrival, ToA.**

I. Introduction

ESTIMATING the location of an event using its times of arrival (ToAs) at a number of sensors is a canonical problem with many applications such as environmental monitoring (e.g., localizing animals by their sounds) and defence/homeland security (e.g., localizing sources of gunfire or explosions). In this paper, we address the problem of ToA-based source localization in acoustic sensor networks when there are multiple events closely spaced in time. We consider a minimalistic model where each sensor has a list of ToAs but the association between ToAs and events is not known *a priori*. The central difficulty is the following: because the speed of propagation of sound is relatively low, the propagation delay between an event and a sensor could vary significantly with the event location. Therefore, when many events occur close to one another in time, the order of ToAs at a sensor could differ from the order of occurrence of the corresponding events. We provide an example in Fig. 1. In such cases, the simple strategy of sorting the ToAs at each sensor in ascending order, associating the ith ToA at each sensor to a common event and then localizing the event will fail. Thus, in the regime of interest to us, the association problem becomes a fundamental bottleneck, in contrast to much of the literature on ToA-based localization, which assumes the association between ToAs and events as given. An additional complication is that the number of ToAs at each sensor may not be the same because of missed and outlier observations, thereby forcing us to estimate the number of events that occurred as well.

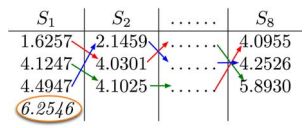

Fig. 1. Three events, that we call "Blue (B)", "Green (G)" and "Red (R)", happen close to one another in time and produce ToAs at 8 sensors. The ToAs at each sensor are sorted in ascending order. The red arrows connect the ToAs produced by the red event and so on. Note that the events need not arrive at the sensors in the same order: for example, the order of ToAs at sensor 1 is RGB, whereas it is BRG at sensor 2. The goal of this paper is twofold. We ask (a) under what conditions can we group the ToAs belonging to each event and localize them and (b) how do we do this in a robust fashion with low complexity? Note that we can have outlier ToAs at some sensors, such as the ToA in the orange bubble at sensor 1, which must be discarded. Additionally, some sensors might miss an event and not have a corresponding ToA, but this is not shown in the figure.

A naïve approach to the association problem is to consider all possible combinations, to use a conventional ToA-based localization algorithm to check whether there is an event location and time of occurrence which can explain each combination of ToAs, and to keep only the "good" explanations. An obvious disadvantage of this approach is its computational complexity: when E events produce ToAs at N sensors, the number of combinations grows exponentially with the number of sensors as E^N. Moreover, there is no guarantee that only E among

Manuscript received November 02, 2011; revised March 14, 2012 and June 01, 2012; accepted June 07, 2012. Date of publication June 25, 2012; date of current version September 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dominic K. C. Ho. This work was supported by the Institute for Collaborative Biotechnologies through Grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

The authors are with the Department of Electrical and Computer Engineering, University of California Santa Barbara, CA 93106 USA (e-mail: sriram@ece.ucsb.edu; madhow@ece.ucsb.edu).

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.2012.2205919

the E^N combinations will be flagged as "valid," and it is unclear how this approach can be extended to handle misses and outliers. An alternative approach [1] is to consider points in space-time as candidate events and quickly discard chunks of space-time where events are not likely to have occurred. However, this algorithm does not account for all the constraints of the problem and is, therefore, prone to estimate more events than the number that occurred. Our goal in this paper is to understand when unambiguous space-time localization of multiple events is possible and to develop techniques for achieving such localization which are robust to sensor non-ideality and have reasonable computational complexity while accounting for all the constraints of the problem.

A. Contributions

We first identify conditions for space-time localization of multiple events based on noiseless ToA readings. We begin with the straightforward observation that, when the temporal separation between any two events exceeds the propagation delay corresponding to the diameter of the deployment region, the association problem is equivalent to sorting the ToAs. We then make precise our intuition that, even when this condition is not satisfied, we should be able to achieve space-time localization if there are "enough" sensors. In particular, we prove that, when the ToAs are noiseless, nine or more sensors suffice to localize two events arbitrarily located in space and time, as long as the sensor deployment is not degenerate (the sensors should not lie on a branch of a hyperbola). We also provide an example showing that six sensors are not enough for unambiguous localization: two different sets of events can provide the same set of ToAs. In simulations, we have found that eight sensors suffice to provide reliable space-time localization even when there are three events, hence it remains an open problem to tighten these results.

Next, we develop a robust, low-complexity algorithm for localization with noisy observations, outliers and misses, assuming that we have enough sensors for unambiguous localization. The algorithm has three stages. The first stage is based on the realization (crucial to sidestepping the computational bottleneck in direct approaches to the association problem) that discretization of the potential times of event occurrence enables us to quickly generate a relatively small set of candidate events. For a hypothesized event time t and an observed ToA τ, the event must lie on a circle of radius $c(\tau - t)$ centered at the sensor, where c is the speed of propagation. Intersections of circles at pairs of sensors generates candidate events, many of which are "phantoms" (arising from intersection of circles corresponding to ToAs for different events) and "duplicates" (intersections of circles for different pairs of sensors, corresponding to ToAs for the same event). At the second stage, we refine these candidate events using Bayesian processing of the observations from all sensors, and develop a goodness metric for each candidate which can be utilized to discard a large number of phantoms and to cluster duplicates. This leaves us with an overcomplete *palette* containing the true event estimates and a few phantoms. At the third stage, we exploit the relatively small palette size to solve a variant of the matching problem on a graph to reject phantoms and to pair event estimates with observations; this can be posed as a binary integer program, which we then relax and solve as a linear program. Simulations show that our algorithm provides performance comparable to that of a genie-based algorithm which is given the correct grouping between ToAs and events.

B. Related Work

There is a vast literature on the general problem of source localization, including algorithms that use ToAs [2], [3], AoAs [4], Time Differences of Arrival (TDoAs) [5]–[7], hybrid versions of these (hybrid TDoA-AoA) [8] and wideband processing of recorded signals [9]. The fundamental limits of wideband localization are derived in [10]. These results are used to arrive at the limits of cooperative localization (where the agents to be localized exchange information with one another) in [11] and different aspects of this problem are surveyed in [12]. Reference [13] uses MUSIC-like techniques to estimate the different taps of a multipath channel (and thereby, extract ToAs). The book [14] and the survey paper [15] explore many of these issues in detail and provide a more exhaustive set of references.

Localization using ultra-wideband (UWB) radios has attracted a lot of research (for example, [16] considers UWB ranging). References [17] and[18] derive the fundamental limits of localizing events from ToAs when the UWB signals go through a multipath/Non Line-of-Sight (NLoS) channel and the transmit/receive clocks are non-ideal. They also simulate the performance of practical estimation algorithms in such settings. Reference [19] improves the localization accuracy in UWB settings by exploiting NLoS paths. Subscriber location service in CDMA systems using ToAs has been explored in [20].

However, most prior papers on ToA-based localization consider one event at a time and hence ignore the association problem central to this paper. A notable exception is [1], which does consider localization of multiple events using ToA sensors. Conceptually, this algorithm can be thought of as discretizing space and time, constructing a count function $C(x, y, t)$ given by the number of sensors agreeing on an event occurring at (x, y, t), and estimating event locations as space-time bins where the count exceeds a threshold. While a naïve implementation of this approach incurs complexity increasing with the size of the deployment region, the algorithm in [1] incurs reasonable complexity by using ideas from interval arithmetic to discard regions where an upper bound on the count falls below the threshold. However, since it completely sidesteps the association problem, the number of events that it outputs can differ from the actual number. Furthermore, it produces a general neighborhood of the event locations, rather than the best possible estimate based on the observations. Our algorithm therefore provides improved performance (close to that of genie-based localization) by refining the location estimates and explicitly solving the association problem in its final stage, while incurring complexity comparable to (slightly smaller than) that in [1].

In terms of feasibility of localization, the number of sensors and conditions on the sensor configuration for perfect event localization from TDoA measurements are characterized in [21].

However, these results are for a single event, and do not address the space-time localization problem considered here.

It is interesting to note that the inspiration for our algorithm comes from the center-surround neural response characteristic of mammalian vision [22]. The complex scenes that we perceive are obtained by intersecting such responses and using feedback from higher layers, which is similar in spirit to our algorithm. This paper builds upon the work presented in a conference paper [23] by adding new results and providing a more complete presentation. Specifically, the section on the limits of localizing multiple events (Section III) is entirely new. We provide a formal statement of the linear program which we could not include in [23] owing to space constraints. We also discuss the complexity of the algorithm and explain a more principled approach to set the threshold for discarding phantom events.

C. Outline

We describe the system model in Section II and characterize the limits of localizing multiple events with ideal measurements in Section III. We describe the algorithm in Section IV, with Sections IV-A–IV-C providing the mathematical details behind each of the three stages. Finally, we present simulation results in Section V and our conclusions in Section VI.

II. System Model

We consider N sensors deployed at locations $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N$ within a two-dimensional region $\mathcal{D}$ that we wish to monitor. The sensors are assumed to be synchronized in time (for example, with a Global Positioning System (GPS) receiver at each sensor or by using a synchronization protocol such as [24]). We observe the system over the time window $[0, T]$. An unknown number of events, E, occur during this period within $\mathcal{D}$. The $e\text{th}$ event is described by the triplet $(\boldsymbol{\varphi}_e, t_e)$ where $\boldsymbol{\varphi}_e \in \mathbb{R}^2$ is the spatial location and t_e the time of occurrence of the event. For any sensor s, the event e is missed with probability p_{miss}, and, with probability $1 - p_{miss}$, produces the following noisy ToA reading:

$$\tau(e \leftrightarrow s) = t_e + \frac{\|\boldsymbol{\varphi}_e - \boldsymbol{\theta}_s\|}{c} + n \tag{1}$$

where c denotes the speed of propagation, $\|.\|$ denotes the two-norm of a vector and n is the measurement noise, assumed to be distributed as $N(0, \sigma^2)$. *In the rest of the paper, we simplify notation by choosing our units so that* $c = 1$. Misses and measurement noise are assumed to be independent across events and sensors. For our statistical processing, we model event occurrence as a space-time Poisson process, with events occurring at rate λ_{LS} per unit time, with locations uniformly distributed over $\mathcal{D}$.

Outliers: Outliers result from "small-scale" events, typically heard at only one sensor (e.g., a nearby slamming car door may trigger an acoustic sensor deployed for detecting far-away explosions), which we are therefore unable to, and not interested in, localizing. We do not model the locations of such events and model their ToAs as arising from a Poisson process with a rate λ_O (per unit time) at each sensor. These processes are assumed to be independent across sensors.

Sensor Observations: Suppose that sensor s records M_s ToAs due to events and outliers in the time window $[0, T]$. We denote the $i\text{th}$ ToA at sensor s by $\tau_s(i)$, where the ToAs are sorted in ascending order at each sensor. Therefore, the set of observations at sensor s is given by $\Omega_s = \{\tau_s(1), \tau_s(2), \ldots, \tau_s(M_s)\}$, with $\tau_s(i) \leq \tau_s(j)$ whenever $i \leq j$. The number of ToAs can vary across sensors because misses and outliers occur independently at each sensor.

III. Feasibility of Localizing Multiple Events

In this section, we investigate the feasibility of localizing multiple events under the most ideal of conditions: no misses, no outliers, no noise. If the order in which the ToAs arrive at each sensor is the same as the order of occurrence of the events, then we can guarantee perfect localization easily: arrange the ToAs in ascending order at each sensor, the $i\text{th}$ largest ToA at each sensor is associated with the $i\text{th}$ event, use a standard single event localization algorithm to localize each event. This approach works when the time interval between any two events is larger than $\frac{D}{c}$, where D is the diameter of the deployment region $\mathcal{D}$ (proof is simple and is omitted). For a network of acoustic sensors deployed over a circular region of radius 1 km, this approach suffices to localize events that are separated by $\frac{(2\times 1000\ \mathrm{m})}{(340\ \frac{\mathrm{m}}{\mathrm{s}})} \approx 6$ s. However, if the time between events is smaller than 6 seconds, we need more sophisticated approaches to the association problem.

When events are closely spaced in time, it is intuitively plausible that we can localize these events correctly if we deploy "enough" sensors. We now show that 9 sensors suffice for localization of two events (with arbitrarily small separation in space and time, for noiseless observations), if the placement of the sensors is not degenerate. Specifically, all sensors should not lie on one branch of a hyperbola. We briefly review the terminology associated with hyperbolas, introduce and define the term *half-hyperbola* and then prove the required result.

A point $\boldsymbol{\theta} \in \mathbb{R}^2$ lies on a hyperbola with foci $\boldsymbol{f}_1, \boldsymbol{f}_2 \in \mathbb{R}^2$ and major axis of length $|a| (a \in \mathbb{R})$ if

$$\Big| \|\boldsymbol{\theta} - \boldsymbol{f}_1\| - \|\boldsymbol{\theta} - \boldsymbol{f}_2\| \Big| = |a|. \tag{2}$$

A hyperbola consists of two branches—the first branch contains the points that lie on $\|\boldsymbol{\theta} - \boldsymbol{f}_1\| - \|\boldsymbol{\theta} - \boldsymbol{f}_2\| = |a|$ and the other branch contains the points that lie on $\|\boldsymbol{\theta} - \boldsymbol{f}_1\| - \|\boldsymbol{\theta} - \boldsymbol{f}_2\| = -|a|$. We use the term *half-hyperbola* to refer to a curve which is one of the branches of a hyperbola, defined as follows:

Definition: A set of points Θ in the two-dimensional plane are said to lie on a half-hyperbola if there exist $\boldsymbol{f}_1,\ \boldsymbol{f}_2 \in \mathbb{R}^2$ and $a \in \mathbb{R}$, so that,

$$\|\boldsymbol{\theta} - \boldsymbol{f}_1\| - \|\boldsymbol{\theta} - \boldsymbol{f}_2\| = a \quad \forall \boldsymbol{\theta} \in \Theta. \tag{3}$$

We are now ready to state our feasibility result.

Theorem 1: Suppose that two events $\mathcal{E}_1 = (\boldsymbol{\varphi}_1, t_1)$ and $\mathcal{E}_2 = (\boldsymbol{\varphi}_2, t_2)$ produce ToAs at each one of N sensors located at $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N$. Then, if the number of sensors $N \geq 9$ and the sensors do not lie on a half-hyperbola, we can guarantee perfect localization of the events.

Proof: We prove the result by contradiction. First, we suppose that an alternate set of events that explains the ToAs at all

the sensors exists, preventing us from localizing the events correctly. We then show that the existence of such an alternate explanation violates one of the conditions in the theorem, thereby proving the required result.

For our ideal observation model (no misses, no outliers, exactly two ToAs at each sensor), we can have an ambiguous reconstruction only if there is an alternate set of events $\mathcal{E}_a = (\boldsymbol{\varphi}_a, t_a)$ and $\mathcal{E}_b = (\boldsymbol{\varphi}_b, t_b)$ which also explain the recorded ToAs.

For the explanation set $\{\mathcal{E}_a, \mathcal{E}_b\}$ to be different from the set of events that produced the ToAs $\{\mathcal{E}_1, \mathcal{E}_2\}$, these sets must differ in at least one event. This can happen in one of two ways:

Case a: Neither of the events $\{\mathcal{E}_a, \mathcal{E}_b\}$ are the same as either of the events $\{\mathcal{E}_1, \mathcal{E}_2\}$.

Case b: One of the events is the same but the other event is different. For example, we might have $\mathcal{E}_a = \mathcal{E}_1$ (meaning, $\boldsymbol{\varphi}_a = \boldsymbol{\varphi}_1$ and $t_a = t_1$), but $\mathcal{E}_b \neq \mathcal{E}_2$ (meaning $\boldsymbol{\varphi}_b \neq \boldsymbol{\varphi}_2$, $t_b \neq t_2$, or both).

We analyze these cases separately.

Case a: All Events Are Different: Since the explanation $\{\mathcal{E}_a, \mathcal{E}_b\}$ produces the same *set* of ToAs as $\{\mathcal{E}_1, \mathcal{E}_2\}$, *exactly* one of the following conditions must be true at each sensor:

- The ToA corresponding to $\mathcal{E}_a$ is equal to the one produced by $\mathcal{E}_1$ and the ToA corresponding to $\mathcal{E}_b$ is equal to that produced by $\mathcal{E}_2$.
- The ToA corresponding to $\mathcal{E}_a$ is equal to the one produced by $\mathcal{E}_2$ and the ToA corresponding to $\mathcal{E}_b$ is equal to that produced by $\mathcal{E}_1$.

Let $\mathcal{W}_1$ denote the subset of sensors which satisfy the first condition and $\mathcal{W}_2$ denote the subset that satisfy the second condition. Then, the total number of sensors $N = |\mathcal{W}_1| + |\mathcal{W}_2|$ where $|\mathcal{A}|$ denotes the cardinality of the set $\mathcal{A}$. We now show that $|\mathcal{W}_1| \leq 4$.

Let $\boldsymbol{\theta}$ denote the location of any sensor in $\mathcal{W}_1$. From the first condition, we see that the location $\boldsymbol{\theta}$ must satisfy both of the following equations:

$$\|\boldsymbol{\theta} - \boldsymbol{\varphi}_a\| - \|\boldsymbol{\theta} - \boldsymbol{\varphi}_1\| = t_1 - t_a, \tag{4}$$

$$\|\boldsymbol{\theta} - \boldsymbol{\varphi}_b\| - \|\boldsymbol{\theta} - \boldsymbol{\varphi}_2\| = t_2 - t_b. \tag{5}$$

Equations (4) and (5) each describe a half-hyperbola. Furthermore, since the events are all different, neither of these equations are trivial. Therefore, any sensor in $\mathcal{W}_1$ must be located at the intersection of two half-hyperbolas. The number of points of intersection of two half-hyperbolas is upper bounded by the number of points in which their "parent" hyperbolas (obtained by including the other branch of each half-hyperbola) intersect. By Bezout's Theorem [21], two hyperbolas intersect in at most 4 points. Thus, we have $|\mathcal{W}_1| \leq 4$. By a similar argument, we can show that $|\mathcal{W}_2| \leq 4$. Thus, to construct two events $\{\mathcal{E}_a, \mathcal{E}_b\}$ so that: (i) they explain the ToAs produced by $\mathcal{E}_1$ and $\mathcal{E}_2$ at all the sensors and (ii) neither $\mathcal{E}_a$ nor $\mathcal{E}_b$ is the same as either of the events $\mathcal{E}_1$ or $\mathcal{E}_2$, we need the number of sensors N to satisfy,

$$N = |\mathcal{W}_1| + |\mathcal{W}_2| \leq 4 + 4 = 8. \tag{6}$$

Case b: One of the Events Is Same: Suppose now that $\mathcal{E}_a = \mathcal{E}_1$ but $\mathcal{E}_b \neq \mathcal{E}_2$. Then, (4) is trivially true and all we can say about sensors in $\mathcal{W}_1$ ($\mathcal{W}_1$ and $\mathcal{W}_2$ have the same definition as in Case (a)) is that their location $\boldsymbol{\theta}$ must satisfy (5).

We now show that the sensors in $\mathcal{W}_2$ will also lie on this half-hyperbola.

Let $\boldsymbol{\theta}$ denote the location of any sensor in $\mathcal{W}_2$. Using the fact that $\mathcal{E}_a = \mathcal{E}_1$ and writing out the conditions that a sensor in $\mathcal{W}_2$ must satisfy, we have

$$\|\boldsymbol{\theta} - \boldsymbol{\varphi}_2\| - \|\boldsymbol{\theta} - \boldsymbol{\varphi}_1\| = t_1 - t_2, \tag{7}$$

$$\|\boldsymbol{\theta} - \boldsymbol{\varphi}_b\| - \|\boldsymbol{\theta} - \boldsymbol{\varphi}_1\| = t_1 - t_b. \tag{8}$$

Subtracting (7) from (8), we see that the location $\boldsymbol{\theta}$ of a sensor in $\mathcal{W}_2$ must satisfy (5). Thus, if we construct an alternate explanation for the recorded data which differs in only one of the two events, then all sensors must lie on a half-hyperbola.

Therefore, if we can find two sets of events $\{\mathcal{E}_1, \mathcal{E}_2\}$ and $\{\mathcal{E}_a, \mathcal{E}_b\}$ that explain the recorded ToAs at N sensors, then either (i) $N \leq 8$ or (ii) all the sensors lie on a half-hyperbola. The true events $\{\mathcal{E}_1, \mathcal{E}_2\}$ obviously explain the recorded data. Since we are given that $N \geq 9$ and we cannot draw a half-hyperbola through all the sensors, an alternate explanation of the data such as $\{\mathcal{E}_a, \mathcal{E}_b\}$ cannot exist (since such an explanation would violate either (i) or (ii)). Consequently, the only explanation for the ToAs are the events $\{\mathcal{E}_1, \mathcal{E}_2\}$ themselves, thereby guaranteeing perfect localization. ■

Remark: Degenerate placement of sensors on a half-hyperbola is a zero probability event for random sensor deployment, and is not possible for regular grid-like deployments.

The immediate question that arises is: can two distinct event sets $\{\mathcal{E}_1, \mathcal{E}_2\}$ and $\{\mathcal{E}_a, \mathcal{E}_b\}$ produce the same ToAs at $N = 8$ sensors (which do not lie on a half-hyperbola), so that these sets cannot be disambiguated? While we are not able to answer this question conclusively, we use insights from the analysis in Case (a) to construct an example where 6 sensors are unable to distinguish between the event sets $\{\mathcal{E}_1, \mathcal{E}_2\}$ and $\{\mathcal{E}_a, \mathcal{E}_b\}$.

Example: We use two ideas in constructing this example: (a) We can choose the event locations $\boldsymbol{\varphi}_1$, $\boldsymbol{\varphi}_2$, $\boldsymbol{\varphi}_a$ and $\boldsymbol{\varphi}_b$ and the parameters $t_1 - t_a$ and $t_2 - t_b$ so that the half-hyperbolas in (4) and (5) intersect in four points. This gives us four sensors in $\mathcal{W}_1$ which cannot disambiguate between the event sets $\{\mathcal{E}_1, \mathcal{E}_2\}$ and $\{\mathcal{E}_a, \mathcal{E}_b\}$. (b) Having made these choices, we show that there is only one free parameter that fixes the two defining half-hyperbolas for any sensor in $\mathcal{W}_2$. We choose this parameter to ensure that these half-hyperbolas intersect in two points.

The example is shown in Fig. 2 and we now provide the details of the construction. We denote the half-hyperbolas that define the set $\mathcal{W}_1$ by $HH_1(4)$ and $HH_2(5)$ respectively. First, we choose $\boldsymbol{\varphi}_1 = (1, 0)$, $\boldsymbol{\varphi}_a = (-1, 0)$ and $t_1 - t_a = 1.3$ to generate HH_1. Next, we set $\boldsymbol{\varphi}_2 = (3, 3.2), \boldsymbol{\varphi}_b = (3, 6)$ and $t_2 - t_b = 2.5$ so that HH_2's axis is at right angles to HH_1's axis (for convenience) and they intersect in four points. Now consider the possible sensor locations in the set $\mathcal{W}_2$. By the definition of this set, the location $\boldsymbol{\theta}$ of any sensor in $\mathcal{W}_2$ must satisfy both the equations $\|\boldsymbol{\theta} - \boldsymbol{\varphi}_2\| - \|\boldsymbol{\theta} - \boldsymbol{\varphi}_a\| = t_a - t_2$ and $\|\boldsymbol{\theta} - \boldsymbol{\varphi}_1\| - \|\boldsymbol{\theta} - \boldsymbol{\varphi}_b\| = t_b - t_1$. Since we have already chosen the event locations, we only need to specify $t_b - t_1$ and $t_a - t_2$ in these equations to construct $\mathcal{W}_2$. However, we cannot choose $t_b - t_1$ and $t_a - t_2$ independently because we have already set

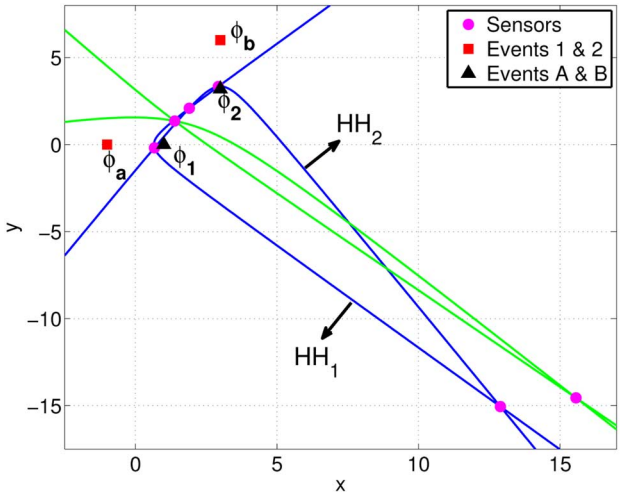

Fig. 2. Each one of the sensors shown by the pink dots record two ToAs—one from Event 1 and the other from Event 2, whose locations are shown by the black triangles. However, events A and B, shown by the red squares, also produce the same set of ToAs at all the sensors. Therefore, the sensors are unable to decide which of the event sets $\{\mathcal{E}_a, \mathcal{E}_b\}$ and $\{\mathcal{E}_1, \mathcal{E}_2\}$ occurred.

$t_1 - t_a = 1.3$ and $t_2 - t_b = 2.5$ and these four quantities add up to zero. Therefore, we choose $t_a - t_2 = 0.3$ (set $t_b - t_1$ appropriately) so that the half-hyperbolas intersect in two points. The six sensors shown in Fig. 2 cannot distinguish between the event pairs $\{(\boldsymbol{\varphi}_a, t_a), (\boldsymbol{\varphi}_b, t_b)\}$ and $\{(\boldsymbol{\varphi}_1, t_1), (\boldsymbol{\varphi}_2, t_2)\}$.

IV. Algorithm Description

We now explain our algorithm for localizing multiple events, assuming that we have deployed enough sensors. The key idea is to narrow down the set of candidate event locations and times to a small set quickly, which we do by discretizing the times at which events occur. While this process generates estimates of the events that occurred, it also generates "phantom" estimates—events that never occurred. We discard "obvious" phantoms by developing a statistical metric and discarding events whose metric is too low. This process is described in Stage 1 (Section IV-A). We then refine the estimates using measurements at all the sensors in Stage 2 (Section IV-B). At this point, we have a palette containing refined estimates of the true events and non-obvious phantoms. In Stage 3, we pick the subset of events from the palette that best explains the observations at the sensors and thereby reject the non-obvious phantoms too. We do this using a linear program and we describe it in Section IV-C.

A. Stage 1: Generating Candidate Events

We begin by hypothesizing that the ith ToA at sensor s and the jth ToA at sensor s' are produced by the same event that occurred at time u. If the measurements are noiseless, we see from (1) that the location of the hypothesized event, denoted by $\mathbf{r}$, must satisfy both the equations

$$u + \|\mathbf{r} - \boldsymbol{\theta}_s\| = \tau_s(i) \Rightarrow \|\mathbf{r} - \boldsymbol{\theta}_s\| = \tau_s(i) - u \quad (9)$$

$$u + \|\mathbf{r} - \boldsymbol{\theta}_{s'}\| = \tau_{s'}(j) \Rightarrow \|\mathbf{r} - \boldsymbol{\theta}_{s'}\| = \tau_{s'}(j) - u. \quad (10)$$

Thus, the hypothesized event must be located at the intersection of two circles centered at $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_{s'}$ with radii $\tau_s(i) - u$ and $\tau_{s'}(j) - u$ respectively. The points of intersection, denoted by $\mathbf{r}_+$ and $\mathbf{r}_-$, can be computed easily in closed-form as

$$\mathbf{r}_\pm = \frac{\boldsymbol{\theta}_s + \boldsymbol{\theta}_{s'}}{2} - \frac{R_{s'}^2 - R_s^2}{2d_{ss'}^2}(\boldsymbol{\theta}_{s'} - \boldsymbol{\theta}_s) \pm \frac{b}{2d_{ss'}^2}(\boldsymbol{\theta}_{s'}^\perp - \boldsymbol{\theta}_s^\perp), \quad (11)$$

where $R_s = \tau_s(i) - u$ and $R_{s'} = \tau_{s'}(j) - u$ are the radii of the circles, $d_{ss'} = \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s'}\|$ is the distance between the sensors, $b = \sqrt{[(R_s + R_{s'})^2 - d_{ss'}^2]\,[d_{ss'}^2 - (R_s - R_{s'})^2]}$ and $\boldsymbol{\theta}_{s'}^\perp - \boldsymbol{\theta}_s^\perp = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (\boldsymbol{\theta}_{s'} - \boldsymbol{\theta}_s)$ is a vector perpendicular to the line joining the sensors $\boldsymbol{\theta}_{s'} - \boldsymbol{\theta}_s$. Additionally, if the measurement noises are small, we can show that the error in the estimated location $\mathbf{e}_\pm = \mathbf{K}_\pm \begin{bmatrix} n_s(i) \\ n_{s'}(j) \end{bmatrix}$ (Appendix A gives an explicit formula for $\mathbf{K}_\pm$) and therefore, the covariance of the estimate is $\mathbf{C}_\pm = \mathbb{E}\left(\mathbf{e}_\pm \mathbf{e}_\pm^T\right) = \sigma^2 \mathbf{K}_\pm \mathbf{K}_\pm^T$. Thus, given an event time u, we can generate candidate event locations and their covariances quickly by intersecting circles drawn at different pairs of sensors.

Fig. 3 illustrates what happens in this process by considering the ToAs produced by two events "red" and "green" at sensors s and s'. The red event generates ToAs $\tau_s(1)$ and $\tau_{s'}(2)$ at sensors s and s'; similarly, the green event produces ToAs $\tau_s(2)$ and $\tau_{s'}(1)$. For each of these ToAs, we hypothesize that it was generated by an event at time u and draw the circles on which such an event must lie. Suppose that the time at which the red event occurred is equal to the hypothesized time u. Then, one of the points of intersection of the red circles will be close to the true event location and this is shown in Fig. 3 by $\hat{\mathcal{E}}$. The red circles also intersect at a second point p_1; however, no event occurred there in reality and hence we call it a *phantom estimate*. Such phantom estimates arise because we have only taken measurements at a pair of sensors (s, s') into account while generating them. We will later use measurements from the other sensors to discard them. Phantom estimates are also generated when we intersect circles drawn using ToAs generated by completely different events (for example, intersecting green and red circles produces the phantoms p_2, p_3, p_6 and p_7) or when we intersect ToAs produced by the same event but the hypothesized time is wrong (for example, the time at which the green event occurred might be different from the hypothesized time u. In this case, intersecting the green circles produces phantoms p_4 and p_5).

We generate a set of candidate events $\{\mathbf{r}_n, u_n, \sigma^2 \mathbf{K}_n \mathbf{K}_n^T\}, n = 1, 2, \ldots$ by repeating this process for different choices of the hypothesized event time u, pair of sensors (s, s') and ToAs at these sensors. Specifically, we do the following:

- Hypothesize a discrete set of event times $\Lambda_{event} = \{\ldots, -2\epsilon, -\epsilon, 0, \epsilon, 2\epsilon, \ldots\}$.
- Pick P pairs of sensors $\Lambda_{sensors} = \{(s_1, s_1'), (s_2, s_2'), \ldots, (s_P, s_P')\}$ randomly (we will explain how to choose P later).
- For $u \in \Lambda_{event}$, $(s, s') \in \Lambda_{sensors}$, $1 \le i \le M_s$, $1 \le j \le M_{s'}$
  - — Intersect circles centered at $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_{s'}$ with radii $\tau_s(i) - u$ and $\tau_{s'}(j) - u$.
  - — Add points of intersection (if any) and their covariances to the list of candidate events.

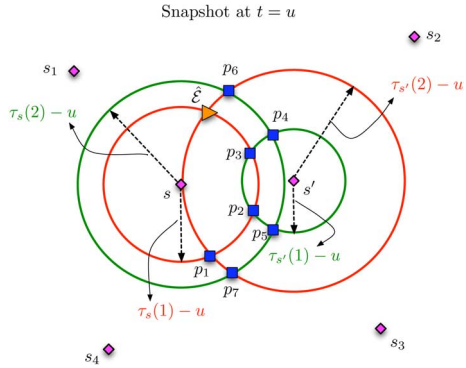

Fig. 3. Geometry of the processing in Stage 1. Six sensors $s, s', s_1, s_2, s_3, s_4$ are shown. There are two events "red" and "green" that produce ToAs at these sensors. The red event produces ToAs $\{\tau_s(1), \tau_{s'}(2)\}$ at sensors s and s'; similarly, the green event produces ToAs $\{\tau_s(2), \tau_{s'}(1)\}$. We hypothesize an event time u and draw circles corresponding to the different ToAs. The time at which the red event occurred is close to u and hence we obtain an estimate of the red event location $\hat{\mathcal{E}}$. However, we also generate phantom estimates $p_1 - p_7$: locations where no event happened in reality.

In addition to phantoms, this process of generating candidates also produces duplicate estimates of true events. To see this, suppose that for a hypothesized event time u, we intersect circles of radii $\tau_s(i) - u$ and $\tau_{s'}(j) - u$ centered at sensors s and s' to produce an event estimate $\mathbf{r}_a$. For the next hypothesized time $u + \epsilon$, intersecting circles of radii $\tau_s(i) - u - \epsilon$ and $\tau_{s'}(j) - u - \epsilon$ centered at these sensors will result in an estimate $\mathbf{r}_b$ that is very close to $\mathbf{r}_a$ (assuming ϵ is small). It is clear that $(\mathbf{r}_a, u)$ and $(\mathbf{r}_b, u + \epsilon)$ are duplicate estimates of the same event and it is sufficient to retain the "better" estimate among the two.

The goal of the rest of the algorithm is to use measurements at the other sensors to (a) discard the phantoms, (b) cluster the duplicate estimates into a single estimate and (c) refine true event estimates to reduce the impact of measurement noise. In the rest of this section, we describe a method to discard "obvious" phantoms and cluster duplicates. We first develop a *goodness* metric for each candidate—the higher its value, the less likely that it is a phantom. Candidates with very low goodnesses are discarded as obvious phantoms. This is described in Section IV-A-I. We then introduce the concept of a *grouping* to cluster duplicates in a principled manner in Section IV-A-II.

1) Discarding Obvious Phantoms Using Goodness: Consider a candidate event $\mathcal{E} = (\mathbf{r}, u)$ with a covariance matrix $\mathbf{C} = \sigma^2 \mathbf{K}\mathbf{K}^T$. We compute the goodness for $\mathcal{E}$ in two steps: first, we calculate individual goodnesses for $\mathcal{E}$ at each sensor and then, multiply them out to obtain an overall goodness. The goodness for event $\mathcal{E}$ at sensor s depends on the difference between the predicted ToA for $\mathcal{E}$ at sensor s, given by $\eta_s = u + \|\mathbf{r} - \boldsymbol{\theta}_s\|$, and the closest among the observed ToAs to the predicted value, denoted by $\tau_s(\mathcal{E})$ $\left(\tau_s(\mathcal{E}) = \arg\min_i |\tau_s(i) - \eta_s|\right)$. Loosely, $\tau_s(\mathcal{E})$ is the "best evidence" that the sensor s has to offer for the event $\mathcal{E}$ having taken place. We define the mismatch to be $z_s = \tau_s(\mathcal{E}) - \eta_s$ and we denote the goodness by $L(z_s)$. Intuitively, we expect the goodness to be large when the mismatch z_s is small and decrease monotonically as z_s increases.

We compute $L(z_s)$ by first conditioning on the event $\mathcal{E}$ being heard or missed at sensor s and obtain the conditional likelihoods $L(z_s|\mathcal{E}\text{ heard})$ and $L(z_s|\mathcal{E}\text{ missed})$. The overall goodness is a weighted average of the conditional likelihoods, with the weights depending on p_{miss} (typically, we choose it to be 5%):

$$L(z_s) = (1 - p_{miss})L(z_s|\mathcal{E}\text{ heard}) + p_{miss}L(z_s|\mathcal{E}\text{ missed}). \tag{12}$$

When an event $\mathcal{E} = (\mathbf{r}, u)$ is heard at sensor s, the mismatch between the predicted ToA η_s and the observed ToA $\tau_s(\mathcal{E})$ arises for two reasons: (a) errors in the spatial estimate of the event $\mathbf{r}$ (obtained by intersecting circles) cause the predicted ToA η_s to be slightly wrong and (b) measurement noise corrupts the observed ToA $\tau_s(\mathcal{E})$. Using these observations, we show in Appendix B that $z_s \sim N(0, \sigma_s^2)$, where

$$\sigma_s^2 = \sigma^2 \left(1 + \frac{\|(\mathbf{r} - \boldsymbol{\theta}_s)^T \mathbf{K}\|^2}{\|\mathbf{r} - \boldsymbol{\theta}_s\|^2}\right). \tag{13}$$

Therefore, we have

$$L(z_s|\mathcal{E}\text{ heard}) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{z_s^2}{2\sigma_s^2}\right). \tag{14}$$

When the event $\mathcal{E}$ is missed at sensor s, the mismatch z_s is obtained by taking the difference between the predicted ToA for $\mathcal{E}$, denoted by η_s, and an observed ToA $\tau_s(\mathcal{E})$. The ToA $\tau_s(\mathcal{E})$ has two properties: (a) it is produced by a *completely different event* and (b) it is the closest among all ToAs at this sensor to η_s. Assuming that the ToAs arrive at the sensors as a Poisson process with rate $\lambda = \lambda_{LS}(1 - p_{miss}) + \lambda_O$, we can show that (details in Appendix B)

$$L(z_s|\mathcal{E}\text{ missed}) = \lambda \exp(-2\lambda|z_s|) \quad \forall z_s. \tag{15}$$

Substituting (14) and (15) in (12), the goodness at sensor s for event $\mathcal{E}$ is given by

$$L(z_s) = \frac{(1 - p_{miss})}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{z_s^2}{2\sigma_s^2}\right) + p_{miss}\lambda \exp(-2\lambda|z_s|). \tag{16}$$

The exponential in the second term decays much slower than the first—it goes down only as $\exp(-|z_s|)$ unlike the first which decays as $\exp(-|z_s|^2)$. Furthermore, in the regime of interest to us, the constants controlling the decay rates λ and σ satisfy $\frac{1}{\lambda} \gg \sigma$. This slows down the decay in the second term relative to the first even further. Therefore, we can neglect the decay in the second term and approximate $L(z_s)$ as

$$L(z_s) \approx \frac{(1 - p_{miss})}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{z_s^2}{2\sigma_s^2}\right) + p_{miss}\lambda. \tag{17}$$

Technically, the mismatches z_s are not independent of each other because the error in the spatial estimate causes a correlated pattern of errors in the predicted ToAs $\{\eta_s\}$ at the different sensors. We ignore such dependence among the mismatch values z_s (this is reasonable when the spatial estimation error is not too large) and compute the overall goodness for the event $\mathcal{E}$, denoted by g, to be

$$g = \sum_{s=1}^{N} \log\left(L(z_s)\right). \tag{18}$$

If the goodness g falls below a threshold κ, we declare the event to be a phantom and discard it from the candidate list. We choose

κ conservatively to ensure that, in the process of discarding phantoms, we do not discard the estimates close to a true event. Specifically, we pick κ so that the estimate of a true event is thrown away with a very small probability δ_{throw}. The threshold is a function of $(N, \sigma^2, p_{miss}, \lambda)$ and we explain how to compute it in Appendix C.

2) Clustering Duplicates via ToA Groupings: We now describe how we cluster duplicate event estimates by introducing the concept of a grouping. Consider an event $\mathcal{E} = (\mathbf{r}, u)$ whose goodness is greater than κ, so that it survives the pruning process described above. We continue to denote the predicted ToA for $\mathcal{E}$ at sensor s by $\eta_s = u + \|\mathbf{r} - \boldsymbol{\theta}_s\|$ and the corresponding "best fit" evidence by $\tau_s(\mathcal{E})$. The grouping $\mathbf{p}$ associated with an event $\mathcal{E}$ is a set of N quantities $\{p_1, p_2, \ldots, p_N\}$ where p_s stores the evidence $\tau_s(\mathcal{E})$ if it is "compelling"; otherwise, it assumes that sensor s has missed the event. Specifically, if the difference $|\tau_s(\mathcal{E}) - \eta_s|$ is smaller than a threshold γ, we set $p_s = \tau_s(\mathcal{E})$; otherwise, we store the string $miss$ in p_s. Duplicated events, such as $(\mathbf{r}_a, u)$ and $(\mathbf{r}_b, u + \epsilon)$ in the example described above, are likely to have the same grouping—since the events are close to one another in space and time, evidence that is "compelling" for one is also likely to be compelling for the other. This observation provides us with a simple rule to cluster events and pick a representative: if two events $(\mathbf{r}_1, u_1)$ and $(\mathbf{r}_2, u_2)$ have groupings $\mathbf{p}_1$ and $\mathbf{p}_2$ that are identical, then we only retain the event with the greater goodness (as defined in (18)).

At the end of this process, we have a list of candidate events $\{\mathbf{r}_n, u_n, \sigma^2\mathbf{K}_n\mathbf{K}_n^T\}$, consisting of true event estimates and "non-obvious" phantoms.

B. Stage 2: Refining the Estimates

The event location estimates in Stage 1 are produced by intersecting circles whose radii are derived from the observed ToAs at a pair of sensors s, s'. Since the measurements are noisy, estimates based on ToAs at only two sensors can have significant errors. In this section, we use the measurements at other sensors to refine such noisy estimates.

Let $\mathcal{E} = (\mathbf{r}, u)$ be an event in the candidate list at the end of stage 1. We only use the sensors that have "compelling" evidence for $\mathcal{E}$ in the refinement process. Specifically, we use sensor s in the refinement process only if the predicted ToA η_s and the corresponding "best fit" ToA $\tau_s(\mathcal{E})$ satisfy $|\tau_s(\mathcal{E}) - \eta_s| \leq \gamma$.

Refinement Procedure: Suppose that the sensors $s_1, s_2, \ldots, s_Q$ have ToAs that are within γ of the predicted ToA for $\mathcal{E} = (\mathbf{r}, u)$ at these sensors. We denote the best evidence for $\mathcal{E}$ at these sensors by $\tau_{s_1}(\mathcal{E}), \tau_{s_2}(\mathcal{E}), \ldots, \tau_{s_Q}(\mathcal{E})$ respectively. Since the refined estimate, denoted by $(\mathbf{r} + \Delta\mathbf{r}, u + \Delta u)$, must fit the measurement model, we have

$$\tau_{s_j}(\mathcal{E}) = u + \Delta u + \|\mathbf{r} + \Delta\mathbf{r} - \boldsymbol{\theta}_{s_j}\| + n_{s_j}, \qquad (19)$$

where $n_{s_j} \sim N(0, \sigma^2)$ and $j = 1, 2, \ldots Q$. If the spatial refinement $\|\Delta\mathbf{r}\|$ is much smaller than the distance $\|\mathbf{r} - \boldsymbol{\theta}_{s_j}\|$ between the event and sensor s_j, we can expand $\|\mathbf{r} + \Delta\mathbf{r} - \boldsymbol{\theta}_{s_j}\|$ as a Taylor series in $\Delta\mathbf{r}$ to obtain

$$\tau_{s_j}(\mathcal{E}) - u - \|\mathbf{r} - \boldsymbol{\theta}_{s_j}\| = \begin{bmatrix} \frac{\mathbf{r}^T - \boldsymbol{\theta}_{s_j}^T}{\|\mathbf{r} - \boldsymbol{\theta}_{s_j}\|} & 1 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{r} \\ \Delta u \end{bmatrix} + n_{s_j}, \quad (20)$$

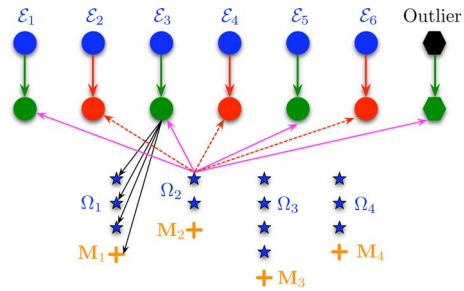

Fig. 4. Modified version of matching problem on a bipartite graph. Events in the overcomplete palette at the output of stage 2 are shown as blue circles in the first row. The observations are denoted by blue stars, with the observations at each sensor arranged in a column. The second of row of nodes show the decisions we make for each event: green circles represent events that we declared to have occurred and red circles denote phantom events. We need to draw edges between the picked events and the observations, subject to constraints, so as to maximize the sum of the values of the edges.

for $j = 1, 2, \ldots, Q$. Let $\mathbf{y}$ denote the Q dimensional vector whose jth component is $\tau_{s_j}(\mathcal{E}) - u - \|\mathbf{r} - \boldsymbol{\theta}_{s_j}\|$ and H denote the $Q \times 3$ matrix whose jth row is $\begin{bmatrix} \frac{\mathbf{r}^T - \boldsymbol{\theta}_{s_j}^T}{\|\mathbf{r} - \boldsymbol{\theta}_{s_j}\|} & 1 \end{bmatrix}$. Then, the least-squares estimate of the refinements $[\Delta\mathbf{r}\ \Delta u]$ is given by

$$\begin{bmatrix} \Delta\hat{\mathbf{r}} \\ \Delta\hat{u} \end{bmatrix} = (H^T H)^{-1} H^T \mathbf{y}. \qquad (21)$$

We update the event location and time estimates and set $\mathbf{r} \leftarrow \mathbf{r} + \Delta\hat{\mathbf{r}}$ and $u \leftarrow u + \Delta\hat{u}$. We now use the refined estimate $(\mathbf{r} + \Delta\hat{\mathbf{r}}, u + \Delta\hat{u})$ as a starting point and repeat the process—this includes computing the grouping for $(\mathbf{r} + \Delta\hat{\mathbf{r}}, u + \Delta\hat{u})$, using the grouping to identify sensors that heard it and then refining the estimate further using the ToAs at these sensors. We typically perform 10 such rounds of refinement for each candidate point from stage 1. The threshold parameter γ—used to decide if a sensor heard/missed $\mathcal{E}$—must be chosen appropriately and from our simulations, we find that choosing $\gamma = 6\sigma$ works well.

After such refinements, we run the clustering algorithm once again to merge duplicates. The output of stage 2 is a palette of candidate events, consisting of true event estimates and non-obvious phantoms, that are refined versions of the estimates from stage 1.

C. Stage 3: Picking True Events From the Palette

The goal of Stage 3 is to find the *subset* of events from the overcomplete palette that best explains (in terms of likelihood) the observations at all sensors. We do this by solving a variant of the matching problem on a bipartite graph, where the events in the palette form one set of nodes and the observations at the sensors form the other set. An example is shown in Fig. 4.

The first set of decisions we need to make are: did event $\mathcal{E}_e, e = 1, 2, \ldots, P$ happen, or is it a phantom? We denote these decisions by the binary variables δ_e: if we declare $\mathcal{E}_e$ to have occurred, we set $\delta_e = 1$; otherwise, we set it to zero. The second row of nodes in Fig. 4 gives an example of such decisions—picked events are shown in green ($\mathcal{E}_1, \mathcal{E}_3$ and $\mathcal{E}_6$) while the ones declared to be phantoms are shown in red ($\mathcal{E}_2, \mathcal{E}_4$ and $\mathcal{E}_5$).

Having decided on a set of events that occurred, we must establish a correspondence between the picked events and the

ToAs at the different sensors (which event produced which ToA at each sensor?). While constructing these correspondences, we must satisfy two types of constraints: event node constraints and observation node constraints.

Event Node Constraints: At each sensor, an event $\mathcal{E}_e$ that we declared to have occurred ($\delta_e = 1$) must either produce one of the observed ToAs or it must be missed. This can be neatly summarized in Fig. 4: exactly one of the edges connecting $\mathcal{E}_3$ to the three ToAs at sensor 1 (blue $\star$) or the miss node M_1 (orange $+$) must be "active". We can capture this with binary decision variables associated with each edge.

Observation Node Constraints: Similarly, a ToA at any sensor must have either been produced by one of the events we declared to have occurred or it must be an outlier. In Fig. 4, if we wish to declare the first ToA at sensor 2 to be an outlier, we activate the edge that joins it to the node marked "Outlier"; on the other hand, if we decide that it was produced by one of the events we declared to have occurred, we activate the appropriate pink edge. Note that we *cannot* activate any of the dotted red edges, because they associate this ToA to events that we have already declared to be phantoms. We now state these correspondence constraints formally.

Let i be a running index for all the observation nodes (including the "miss" nodes) as shown in the figure. Let $\Omega_s^{\#} = \Omega_s \bigcup M_s$—the set of all observed ToAs at sensor s and the miss node M_s—denote the *observation nodes* at sensor s. Let $w_{ie} = 1$ if we activate the edge between observation i and event $\mathcal{E}_e$; otherwise, we set $w_{ie} = 0$.

Event Node Constraints: Consider a picked event $\mathcal{E}_e$ (one with $\delta_e = 1$). Since it must be associated with exactly one observation node at each sensor s, we have

$$\sum_{i \in \Omega_s^{\#}} w_{ie} = 1 \quad \forall s, \{\forall e : \delta_e = 1\}$$
$$w_{ie} \in \{0, 1\}. \tag{22}$$

To convert this into a constraint that is valid for all events—and not just for those that are picked—we can rewrite (22) as

$$\delta_e \left[\left(\sum_{i \in \Omega_s^{\#}} w_{ie} \right) - 1 \right] = 0 \quad \forall s, e. \tag{23}$$

When $\delta_e = 1$, this reduces to the constraint in (22) and when $\delta_e = 0$, the constraint is trivially satisfied. Therefore, we need to choose our decisions w_{ie} and δ_e so that they satisfy the constraints

$$\sum_{i \in \Omega_s^{\#}} w_{ie} \delta_e = \delta_e \quad \forall s, e. \tag{24}$$

Observation Node Constraints: Next, we consider the constraints at a node i that is an observed ToA at one of the sensors (*not* one of the miss nodes). Let $\mu_{iO} = 1$ if we declare observation i to be an outlier; otherwise, we set $\mu_{iO} = 0$. If we wish to associate it with a picked event (say, event e), then we must have $w_{ie}\delta_e = 1$. Since the observation node i must either be paired with a picked event or declared as an outlier, we have

$$\sum_{e=1}^{P} w_{ie}\delta_e + \mu_{iO} = 1 \quad \forall i \in \Omega_1 \cup \Omega_2 \cdots \cup \Omega_N$$
$$w_{ie}, \mu_{iO}, \delta_e \in \{0, 1\} \quad \forall i, e. \tag{25}$$

Cost Function: We pick events from the palette and choose their correspondence with the observed ToAs to maximize the likelihood of the observations given these decisions. Computing the likelihood is simplified by the following fact: given the decisions (picking events and choosing the correspondence), the observations at two sensors s and s' are independent. Thus, we can set the value of each edge individually. Specifically, the value of an edge that joins an observation node to an event node is the log-likelihood of the observation given that the event produced it. Our goal is to pick events and activate edges, subject to the aforementioned constraints, so that sum of the values of the *activated* edges is maximized. We now specify the values of different edges.

The value of an edge between an event node $\mathcal{E}_e$ and the miss node M_s at sensor s is $\log(p_{miss})$. Consider the edge between event $\mathcal{E}_e = (\mathbf{r}_e, u_e)$ and the jth observation at sensor s, $\tau_s(j)$. For $\mathcal{E}_e$ to produce this observation, two things must happen: (a) $\mathcal{E}_e$ must be heard at sensor s (which happens with probability $1 - p_{miss}$ and (b) given that it was heard at sensor s, it must produce the observation $\tau_s(j)$. Therefore, the value of the edge between $\mathcal{E}_e$ and $\tau_s(j)$ is

$$\mathrm{val}(\mathcal{E}_e, \tau_s(j)) = \left(\log \frac{1 - p_{miss}}{\sqrt{2\pi\sigma^2}} \right) \times - \frac{(\tau_s(j) - u_e - \|\mathbf{r}_e - \boldsymbol{\theta}_s\|)^2}{2\sigma^2}. \tag{26}$$

The value of the edge joining an observation at sensor s to the outlier node is trickier to compute. Since the outliers are generated by a Poisson process of rate λ_O, the chance that there are k outliers at sensor s over an observation window of length T is $e^{-\lambda_O T} \frac{(\lambda_O T)^k}{k!}$. The logarithm of this quantity is (ignoring constants),

$$L = k \log(\lambda_O T) - \log(k!). \tag{27}$$

If the term $\log(k!)$ is absent, declaring an observation to be an outlier has the value $\log(\lambda_O T)$ and the overall value of declaring k outliers is $k \log(\lambda_O T)$. However, the presence of the term $\log(k!)$ implies that the value of declaring an observation to be an outlier cannot be a constant quantity, say $\log \alpha$; rather, it also depends on the number of *other* observations we declare to be outliers. To circumvent this problem, we approximate the distribution of the number of outliers to be geometric with parameter q (as opposed to the true Poisson $(\lambda_O T)$ distribution). With this approximation, the log-likelihood of observing k outliers at sensor is $k \log q$. Thus, we can set the value of an edge that joins an observation at sensor s to the outlier node to $\log q$. We choose q to ensure that the probabilities assigned by the Poisson and geometric distributions are close to one another (we minimize the mean-square error between the sequences $\{(1 - q)q^n, n = 0, 1, \ldots\}$ and

$\{e^{-\lambda_O T}\frac{(\lambda_O T)^n}{n!}, n = 0,1,\ldots\}$). The overall value of the decisions $\{\delta_e, w_{ie}, \mu_{iO}\}$ is given by

$$J = \sum_s \sum_{i\in\Omega_s^{\#}} \sum_{e=1}^{P} c_{ie} w_{ie} \delta_e + \sum_s \sum_{i\in\Omega_s} c_{iO}\mu_{iO}, \qquad (28)$$

where c_{ie} is the value of activating the edge between the ith observation node (this could be a miss node) and the eth event and $c_{iO} = \log q$ is the value of declaring the ith observation node to be an outlier (this summation is only over observed ToAs and does not include miss nodes).

Since the decision variables are all binary valued, our problem can be stated as a binary integer program. By massaging the problem, we can convert it to a form where the cost function and the constraints are linear in the decision variables. We provide these details in Appendix D. We relax the integer program and allow the variables to take any value between 0 and 1. This allows us to solve the problem as a linear program (LP), which is much faster. In all our simulations, when the number of sensors is "large enough" (typically, we simulate $N = 8$ or $N = 16$ sensors and $E = 3$ events), we find that, the decision variables that optimize the LP only take the values 0 or 1 and never take any value in between. This is analogous to the efficacy of LP decoding for turbo-like codes and it is of interest to investigate whether the literature in this area [25] can shed light on the performance of our algorithm. Finally, we declare the events "picked" by the LP (those with $\delta_e = 1$) to have taken place.

D. Complexity

We now give some insights into the complexity of the algorithm and begin by analyzing the circle-intersection process. Consider a pair of sensors s and s' with E ToAs each. For each hypothesized event time $u = l\epsilon$, we need to intersect E^2 pairs of circles (every combination of ToAs, one from each sensor) at these sensors. Over a time window T, we repeat this $\frac{T}{\epsilon}$ times and if we repeat the whole process at P pairs of sensors, the total number of circle intersections is $\frac{TE^2P}{\epsilon}$.

For a low-complexity algorithm, we would like to keep P as small as possible. However, if we intersect circles at too few sensors, we run the risk of not generating candidates close to some of the events. Suppose that we are intersecting circles centered at (s, s') and one of the sensors misses the ToA belonging to $\mathcal{E}$. Then, we will not generate any candidate close to $\mathcal{E}$ and this happens with probability $1-(1-p_{miss})^2 \approx 2p_{miss}$. Since the misses are independent across sensors, if we intersect circles drawn at P *distinct* pairs of sensors, the chance of missing any event $\mathcal{E}$ is $(2p_{miss})^P$. We choose P to ensure that this value is small (for example, with $p_{miss} = 0.05$, we choose $P = 6$ so that $(2p_{miss})^P = 10^{-6}$). Since P depends only on p_{miss} and not on the number of sensors N, the complexity of the circle-intersection process does not increase with the number of sensors.

Increasing ϵ is the only way to reduce the complexity of the circle-intersection process. However, we cannot increase ϵ too much for two reasons: while computing the threshold for the goodness, we have ignored the finiteness of ϵ. If ϵ is too large, the estimate of an event might be far away from the event itself, resulting in it being discarded because its goodness falls below the threshold. Secondly, even if we choose the threshold conservatively so that the estimate survives, if it is too far away from the actual event, the iterative refinement process in Stage 2 might fail to bring the estimate closer. For example, we find that with $N = 8$ sensors (and other parameters described in Section V), choosing $\epsilon = 0.16$ s results in a failure rate of 1% and $\epsilon = 0.2$ s has a failure rate of 2% (we declare an estimation failure if the (a) the estimated location for any event is at least 40 m from the true location or (b) the algorithm outputs a wrong number of events). Similarly, we find that the failure rate with $N = 16$ sensors and $\epsilon = 0.2$ s is 1.5% and that with $\epsilon = 0.24$ s is 1.6%. Quantifying such failure rates analytically or providing guidelines for choosing the "largest" possible value of ϵ that avoids such failures are beyond the scope of this paper and are an important topic for future work.

Computing the complexity of the other stages is complicated because we discard many candidates (either when their goodness falls below the threshold or when they get clustered), but it is difficult to quantify the number of discarded candidates. However, given the number of candidates at each step, we can compute the respective complexities and this is shown in Fig. 5 (we provide a detailed pseudocode of each block in Fig. 5 in a technical report [26]). We now describe the *average values* for the number of candidates that we observe at different stages of the simulations.

We generate $E = 3$ events over a time window $T = 5$ s and simulate sensor deployments of $N = 8$ and $N = 16$ sensors. The mean number of candidates generated after the circle intersection process (W_1 in Fig. 5) is virtually the same in both cases: for $N = 8$ sensors, we generate 5777 candidates and for $N = 16$ sensors, we generate 5713 candidates. It is at the next step that we see the value of additional sensors. The number of candidates whose goodness exceeds the threshold (W_2 in Fig. 5) is 879 when $N = 8$ (15% of W_1). However, when the number of sensors increases to $N = 16$, the number of points at this stage is only 66 (only 1.4% of W_1). Similarly, after clustering we have $W_3 = 165$ candidates when $N = 8$ sensors and $W_3 = 38$ candidates when $N = 16$. We see that stage 1 is very effective in reducing the number of candidates (down to 2.87% and 0.67% of W_1 respectively).

The time taken to intersect circles is nearly identical for $N = 8$ and $N = 16$ sensors. However, the time taken for subsequent stages is larger when $N = 8$, since there are more phantom estimates to be discarded. Specifically, when $N = 8$, the time taken by Stages 2 and 3 is 62% of the time taken to intersect circles. However, with $N = 16$, the corresponding fraction is only 13.48%.

V. Simulation Results

Sensor Deployment Model: We run two sets of simulations—one with a "moderate" density of sensors and the other with a larger deployment density. For moderate density, we place $N = 8$ sensors at random in a circular region of radius $R = 1020$ meters. The denser deployment consists of $N = 16$ sensors placed at random over an identical region. Note that the sensor locations do influence the localization error and the properties of optimal sensor placement are studied in [27].

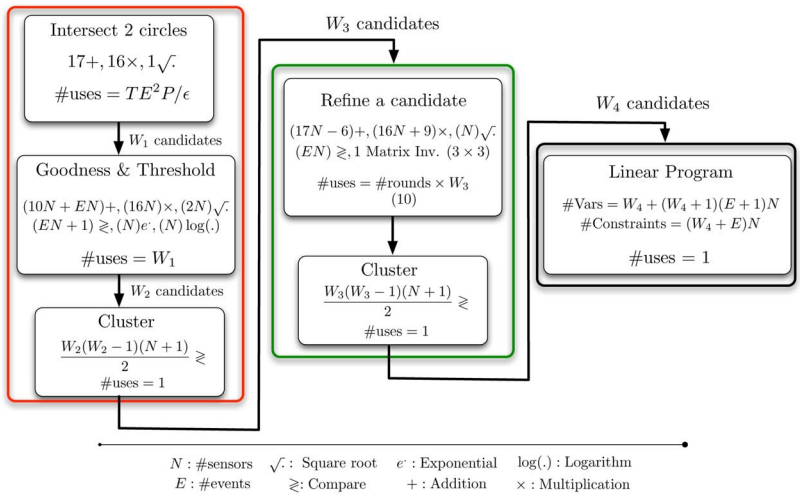

Fig. 5. A schematic sketching the different stages of the algorithm and their complexities. Stages 1, 2, and 3 are shown in the left, middle and right boxes, respectively. A stage is divided into blocks and the block description contains a list of the basic operations it uses and the number of times it is executed (# uses). The number of candidates that survive different blocks are denoted by $W_i, i = 1, 2, 3, 4$.

However, averaging over random deployments facilitates comparison of the proposed scheme with genie-based localization, showing that, even without optimization of sensor locations, the performance of our scheme matches that of a genie in most cases.

Event Generation Model: We generate $E = 3$ events, with their times chosen uniformly at random from the interval 0–5 seconds. We choose the event locations so that they are "inside" the convex hull of the sensors. Specifically, we pick them randomly from a region that is a scaled-down version of the convex hull of the sensors, with the scale-factor being 75%. We generate outliers at a rate $\lambda_O = \frac{3}{100}$ events/sec at each sensor. To compute the goodness, we assume that every sensor observes ToAs arriving at a rate $\lambda = \frac{63}{100} (= \frac{3}{5} + \frac{3}{100})$ events/sec.

Measurement Model: We set the speed of sound c to 340 m/s. Guided by our experimental results in [28], we choose the standard deviation of the measurement noise σ to be 0.02 s and the probability with which the sensors miss an event to be $p_{miss} = 5\%$.

In our processing, we assume that λ and p_{miss} are known (obtained, say, using tests like [28]). When these values are unknown and we approximate them, the algorithm could return a set of events that is larger/smaller set than the truth. For example, when the assumed cost for declaring an outlier is larger than the true cost, the algorithm might group outlier observations at different sensors as an event (even though the event might not fit the observations very well). A detailed analysis of the algorithm's sensitivity to the knowledge of λ and p_{miss} (and robustifying it) is beyond the scope of this paper.

Algorithm Choices: We set the granularity of the hypothesized event times at $\epsilon = 0.08$ s. We choose the probability with which an actual event is discarded at the end of Stage 1 to be $\delta_{throw} = 4 \times 10^{-5}$ and compute the threshold κ accordingly (see (40)). Finally, we choose q—the parameter of the geometric distribution that approximates the Poisson $(\lambda_O T)$ distribution—to be 0.14.

We run 100 trials of the algorithm with the above parameters and benchmark its performance with a "genie" that computes the ML estimate of the events, given the correct association of the ToAs. To do this, we discretize a 70 m × 70 m × $\frac{70}{340}$ s space-time neighborhood of the event $\mathcal{E}_e = (\boldsymbol{\varphi}_e, t_e)$ with a granularity of 1 m along each of the spatial dimensions and 3 ms along the time dimension. We then compute the genie estimate by choosing the point on the discrete grid that best fits the ToAs produced by $\mathcal{E}_e$ at different sensors in the least-squares sense. Figs. 6(a) and 6(b) show scatterplots of the localization errors, with the genie estimate (along x-axis) against the errors with the proposed estimate (along y-axis), for the cases of $N = 8$ and $N = 16$ sensors respectively. We make the following observations from these figures: (a) In all trials, our algorithm correctly estimates the number of events to be three. (b) The localization errors produced by the proposed algorithm and the genie are virtually identical (most points are along the $y = x$ line, shown in red), demonstrating the efficacy of the proposed scheme. The average localization errors for the $N = 8$ and $N = 16$ sensor cases are 5.32 m and 3.69 m respectively.

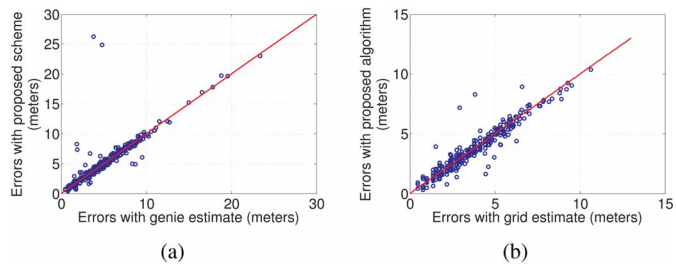

Fig. 6. Scatterplots of localization error with the genie-based scheme (x-axis) against those with the proposed algorithm (y-axis) for $N = 8$ and $N = 16$ sensors. We see that the errors are nearly equal (falling along the $y = x$ line), illustrating the efficacy of the proposed algorithm (a) $N = 8$ sensors (b) $N = 16$ sensors.

Comparison With Algorithm in [1]: For comparison, we simulated the algorithm in [1] with the same scenarios. The algorithm uses two parameters: ρ is the maximum difference between the predicted and observed ToAs for a sensor to "agree" on a candidate event and C_{thresh} is the number of sensors that

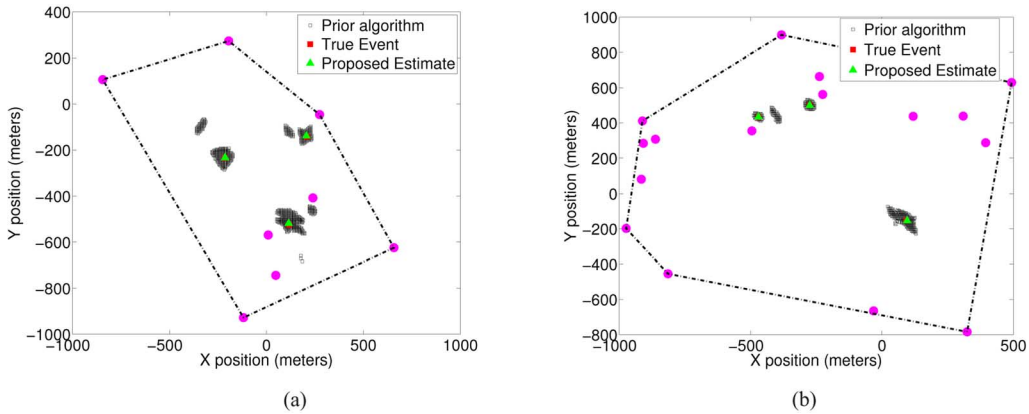

Fig. 7. Comparing the proposed algorithm and [1] using sample runs with $N = 8$ and $N = 16$ sensors. The sensors are represented by circles, and the three event locations are shown using squares. The event estimates produced by [1] are large blobs surrounding the events, and we see that there are more blobs than the number of events. In contrast, the proposed algorithm produces three event estimates (triangles) that virtually coincide with the true event locations (squares).

need to agree on a candidate for it to survive. We set $\rho = 0.04$ s, $C_{thresh} = 6$ for when the number of sensors $N = 8$ and $C_{thresh} = 13$ when $N = 16$ sensors. We chose the tolerance ρ to be as small as possible, while ensuring that an actual event at (x, y, t) is not discarded by a sensor because of large measurement noise (with high probability). Similarly, we chose C_{thresh} to ensure that the chance of the algorithm failing to output an estimate for an event that occurred is small ($\approx 0.006 - 0.007$). The results from a sample run with $N = 8$ sensors are plotted in Fig. 7(a). The estimates produced by [1] are not precise points, but regions surrounding the true events (shown in red squares). These regions can be fairly large—in the blob around the event at $(-213.6, -233.1)$, the farthest point is roughly 70 m from the event itself. Secondly, we see that the number of blobs (six) are more than the number of events (three), indicating that some of them are phantoms. In contrast, the proposed algorithm returns exactly three estimates (shown in green triangles) that are very close to the true event locations. This is because the clustering and linear refinement procedures coalesce the blob into a single point close to the event and the linear program rejects phantoms.

As the number of sensors increase (keeping the number of events and the time window fixed), we observe that the number of phantom blobs and the sizes of the blobs typically decrease. This is expected: a larger set of measurements provide the algorithm in [1] with increased resolvability, even though it does not exploit all the constraints. However, on some trials we do see phantom blobs even with $N = 16$ sensors: an example is shown in Fig. 7(b) (the blob around $(-404,452)$ is a phantom).

The runtimes for the proposed algorithm and [1] were comparable. For example, with $N = 8$ sensors, the mean time taken by the proposed algorithm was 1.16 s, while the time taken by [1] was 2.5 s. As the number of sensors increased to $N = 16$, the times taken were virtually identical: 0.75 s for the proposed algorithm and 0.74 s for the algorithm in [1]. These numbers are heavily dependent on the software used and the implementation, hence we only provide them as an indicator that the complexities are comparable. An interesting question (beyond our current scope, but of interest for future work) is whether the ideas from [1] could be combined with ours to speed up Stage 1 (which generates an overcomplete set of events), while keeping the refinement and linear programming stages of our algorithm.

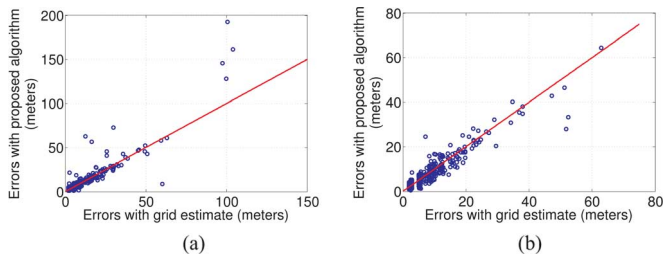

Fig. 8. Scatterplots of localization error with the genie-based scheme (x-axis) against those with the proposed algorithm (y-axis) for $N = 8$ and $N = 16$ sensors, with the events lying outside the convex hull of the sensors. The errors are nearly equal (falling along the $y = x$ line) (a) $N = 8$ sensors (b) $N = 16$ sensors.

Events Outside the Convex Hull: We consider random deployments of $N = 8$ and $N = 16$ sensors in a region of radius $R = 510$ m. The events are constrained to lie outside the convex hull, but within a scaled version of the convex hull (with scale factor 130%). To compute the genie estimate, we discretize a $(200 \text{ m} \times 200 \text{ m} \times \frac{200}{340} \text{ s})$ region around the event with a granularity of 3.4 m along the spatial dimensions and 0.01 s along the time dimension. All other parameters are held the same. The number of events is estimated correctly. We show scatterplots of the errors produced by the genie estimate against the errors produced by our algorithm for $N = 8$ sensors and $N = 16$ sensors in Figs. 8(a) and 8(b) respectively. In most trials, the estimation errors produced by the proposed scheme are virtually the same as those from the genie-based scheme. However, they are both larger than when the events lie within the convex hull of the sensors: the average localization error is 13.77 m with $N = 8$ sensors and 9.46 m with $N = 16$ sensors. In a few trials, the localization error with both the genie and proposed scheme is large (100 m with 8 sensors and 60 m with 16 sensors). This is consistent with prior work showing that the uncertainty in localizing events outside the convex hull can be large (e.g., [29] gives examples where the determinant of the Fisher Information Matrix falls off rapidly outside the convex hull of the sensors).

Thus, in practice, we would want to have a dense enough sensor deployment such that any point of interest does fall within the convex hull of a large enough number of sensors.

VI. Conclusions

The problem of associating ToAs with events is a fundamental bottleneck in localizing multiple events in space and time. Our results on feasibility and ambiguity indicate that space-time localization should be possible with "enough" sensors. We proposed an algorithm that avoids the computational bottlenecks of direct approaches to the association problem. A key element of our approach is to quickly generate a relatively small set of candidate events by discretizing hypothesized event times, and to then apply detection-theoretic criteria for rejecting obvious phantoms. Only after further refinement of the surviving location estimates do we attack the association problem through a linear programming relaxation that matches ToAs with events, and identifies likely misses and outliers. We have used a slightly modified version of this algorithm in an experimental demonstration with much bigger scope; see [30] for a high level overview. The performance of the algorithm was excellent, including on the few occasions when the events were close to one another in time. We omit details owing to lack of space.

There are a number of interesting questions that require further investigation. How do the feasibility and ambiguity results (which consider two events) generalize to more than two events? What are fundamental bounds on the achievable accuracy of space-time localization in the presence of noise? (Existing bounds address spatial localization for a single event.) Finally, how can the approach here be generalized to a network of heterogeneous sensors? For example, a particular ToA observation, in conjunction with a hypothesized event occurrence time, sketches out a circle, an Angle of Arrival sensor with a specified error generates a conical region in space, and a binary proximity sensor [31] generates a circular region. Developing algorithms for integration of such information for source localization and tracking of multiple events is an important topic for future work.

Appendix

A. Spatial Covariance of Circle Intersection Estimates

Assuming that the measurement noises $n_s(i)$ and $n_{s'}(j)$ corrupting the ToAs $\tau_s(i)$ and $\tau_{s'}(j)$ are small, we can use a Taylor's series expansion of (11) to show that the errors in the estimates $\mathbf{e}_\pm$ are given by $\mathbf{e}_\pm = \mathbf{K}_\pm \begin{bmatrix} n_s(i) \\ n_{s'}(j) \end{bmatrix}$ where,

$$\mathbf{K}_\pm = \mathbf{M} \begin{bmatrix} \frac{R_s}{d_{ss'}} & \frac{-R_{s'}}{d_{ss'}} \\ \frac{R_s(d_{ss'}-\beta)}{d_{ss'}\sqrt{R_s^2-\beta^2}} q_\pm & \frac{\beta R_{s'}}{d_{ss'}\sqrt{R_s^2-\beta^2}} q_\pm \end{bmatrix} \tag{29}$$

with $\mathbf{M} = \begin{bmatrix} \frac{\boldsymbol{\theta}_{s'}-\boldsymbol{\theta}_s}{d_{ss'}} & \frac{\boldsymbol{\theta}_{s'}^\perp-\boldsymbol{\theta}_s^\perp}{d_{ss'}} \end{bmatrix}$, $\beta = \frac{(R_s^2-R_{s'}^2+d_{ss'}^2)}{2d_{ss'}}$ and $q_\pm = \operatorname{sign}\left((\mathbf{r}_\pm - \boldsymbol{\theta}_s)^T(\boldsymbol{\theta}_{s'}^\perp - \boldsymbol{\theta}_s^\perp)\right)$.

B. Computing $L(z_s|\mathcal{E}\ heard)$ and $L(z_s|\mathcal{E}\ missed)$

Assuming that the error in the estimated event location $\|\mathbf{e}\|$ is much smaller than the distance between the event and sensor s $\|\mathbf{r}-\boldsymbol{\theta}_s\|$, we can expand $\|\mathbf{r}+\mathbf{e}-\boldsymbol{\theta}_s\|$ as a Taylor series in $\mathbf{e}$ and retain only the linear term, to approximate it as

$$\|\mathbf{r}+\mathbf{e}-\boldsymbol{\theta}_s\| \approx \|\mathbf{r}-\boldsymbol{\theta}_s\| + \left\langle \mathbf{e}, \frac{\mathbf{r}-\boldsymbol{\theta}_s}{\|\mathbf{r}-\boldsymbol{\theta}_s\|} \right\rangle, \tag{30}$$

where $\langle .,. \rangle$ denotes the standard inner product. Using this approximation in the definition of $\tau_s(\mathcal{E})$, we get

$$\tau_s(\mathcal{E}) \approx u + \|\mathbf{r}-\boldsymbol{\theta}_s\| + \left\langle \mathbf{e}, \frac{\mathbf{r}-\boldsymbol{\theta}_s}{\|\mathbf{r}-\boldsymbol{\theta}_s\|} \right\rangle + n. \tag{31}$$

Recognizing $u + \|\mathbf{r}-\boldsymbol{\theta}_s\|$ to be the predicted ToA η_s and using the fact that $\tau_s(\mathcal{E}) - \eta_s = z_s$, we get

$$z_s = \left\langle \mathbf{e}, \frac{\mathbf{r}-\boldsymbol{\theta}_s}{\|\mathbf{r}-\boldsymbol{\theta}_s\|} \right\rangle + n. \tag{32}$$

From (29), the error in the location estimate $\mathbf{e}$ can be expressed as $\mathbf{K}\begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$ where n_1 and n_2 are independent $N(0,\sigma^2)$ random variables. Since n_1, n_2 and n are all Gaussian random variables with zero mean, z_s is also a zero mean Gaussian random variable whose variance, denoted by σ_s^2, is given by

$$\sigma_s^2 = \sigma^2\left(1 + \frac{\|(\mathbf{r}-\boldsymbol{\theta}_s)^T\mathbf{K}\|^2}{\|\mathbf{r}-\boldsymbol{\theta}_s\|^2}\right). \tag{33}$$

Thus, we get the expression in (14) for the conditional likelihood $L(z_s|\mathcal{E}\ \text{heard})$.

Now, suppose that event $\mathcal{E}$ is missed at sensor s and $\tau_s(\mathcal{E})$ is the closest among all ToAs recorded at sensor s to the predicted ToA for $\mathcal{E}$, denoted by η_s. Two conditions must be satisfied for this to have happened: (a) there must be no ToAs at sensor s that are closer to η_s than $\tau_s(\mathcal{E})$. In other words, there must be no ToAs in the interval $[\eta_s - |z_s|, \eta_s + |z_s|]$ where $z_s = \tau_s(\mathcal{E}) - \eta_s$. (b) We must observe a ToA close to $\eta_s + z_s$ i.e., there must be a ToA in the infinitesimal interval $[\eta_s + z_s,\ \eta_s + z_s + dz]$. Assuming that each sensor observes ToAs as a Poisson process with rate $\lambda = \lambda_{LS}(1 - p_{miss}) + \lambda_O$, the probability of (a) happening is $\exp(-2\lambda|z_s|)$ and the likelihood of (b) is λdz. Since the time intervals considered in (a) and (b) do not overlap, the events are independent and we obtain

$$L(z_s|\mathcal{E}\ \text{missed})dz = \exp(-2\lambda|z_s|)\lambda dz. \tag{34}$$

Therefore, $L(z_s|\mathcal{E}\ \text{missed}) = \lambda\exp(-2\lambda|z_s|) \quad \forall z_s$.

C. Computing the Threshold κ

Consider the eth event $\mathcal{E}_e = (\boldsymbol{\varphi}_e, t_e)$ and let $\mathcal{X}_0$ and $\mathcal{X}_1$ denote the set of sensors that missed and heard $\mathcal{E}_e$ respectively. When sensor s hears the event $\mathcal{E}_e$, the first term in (17) dominates the second (for the parameter choices we are interested in) and we can approximate $\log L(z_s)$ as

$$\log L(z_s) \approx \left[\log\frac{1-p_{miss}}{\sqrt{2\pi\sigma_s^2}}\right] - \frac{z_s^2}{2\sigma_s^2} \quad s \in \mathcal{X}_1, \tag{35}$$

where $z_s \sim N(0, \sigma_s^2)$. To further simplify the analysis, we neglect variations in the prediction error variance σ_s^2 across sensors *in the first term* of the above expression. We set $\sigma_s^2 \approx \sigma^2$ in the first term of (35) and get

$$\log L(z_s) \approx \left[\log \frac{1-p_{miss}}{\sqrt{2\pi\sigma^2}}\right] - \frac{z_s^2}{2\sigma_s^2} \quad s \in \mathcal{X}_1. \tag{36}$$

At a sensor s' that missed $\mathcal{E}_e$, the first term in (17) dies down rapidly and we can approximate $\log L(z_{s'})$ as

$$\log L(z_{s'}) \approx \log p_{miss}\lambda \quad s' \in \mathcal{X}_0. \tag{37}$$

Therefore, if N_1 sensors hear the event and $N-N_1$ sensors miss the event, we can use (36) and (37) to obtain an approximate expression for the goodness of $\mathcal{E}_e$:

$$g = \left[\log \frac{1-p_{miss}}{\sqrt{2\pi\sigma^2}}\right] N_1 + [\log(p_{miss}\lambda)](N-N_1) - \frac{1}{2}\sum_{s\in\mathcal{X}_1}\frac{z_s^2}{\sigma_s^2}. \tag{38}$$

Since the prediction error $z_s \sim N(0,\sigma_s^2)$, the term $\sum_{s\in\mathcal{X}_1}\frac{z_s^2}{\sigma_s^2}$ in (38) is a chi-squared distributed random variable with N_1 degrees of freedom with a non-zero mean given by the first two terms. We denote this distribution by $f_G(g|N_1)$. By the law of total probability, the unconditional distribution of the goodness $f_G(g)$ is given by

$$f_G(g) = \sum_{N_1=0}^{N} f_G(g|N_1)\, p_{miss}^{N-N_1}\,(1-p_{miss})^{N_1}\binom{N}{N_1}. \tag{39}$$

We set the threshold κ so that the chance that the goodness for $\mathcal{E}_e$ is lower than κ is equal to δ_{throw}:

$$P(G<\kappa) = \int_{-\infty}^{\kappa} f_G(g)dg = \delta_{throw}. \tag{40}$$

We run Monte Carlo simulations to generate samples of G according to (39) and then pick the threshold to satisfy (40).

D. Integer Program Formulation

From (23), (25) and (28), we see that the decision variables w_{ie} are always multiplied by δ_e and never occur by themselves. We can therefore define new decision variables $\mu_{ie} \triangleq w_{ie}\delta_e$ that are also binary-valued. We now pose the problem of maximizing the cost function in (28) subject to the constraints in (23) and (25) as the following binary integer program:

$$\begin{aligned}
&\max J = \sum_s \sum_{i\in\Omega_s^{\#}} \sum_{e=1}^{P} c_{ie}\mu_{ie} + \sum_s \sum_{i\in\Omega_s} c_{iO}\mu_{iO}\\
&\sum_{i\in\Omega_s^{\#}} \mu_{ie} = \delta_e \quad \forall s, e\\
&\sum_{e=1}^{P} \mu_{ie} + \mu_{iO} = 1 \quad \forall i \in \Omega_1 \cup \Omega_2 \cdots \cup \Omega_N\\
&\delta_e, \mu_{ie}, \mu_{iO} \in \{0,1\} \quad \forall\, i, e..
\end{aligned} \tag{41}$$

The variables μ_{ie} and δ_e are also linked through the relationship $\mu_{ie} = w_{ie}\delta_e$, where $w_{ie} \in \{0,1\}$. It might appear that we have omitted these constraints from the above formulation. We now show that these constraints can indeed be excluded without any change to the optimal solution.

Consider a "complete" formulation where we include these constraints. First, we observe that, since the variables w_{ie} do not appear in the objective function, any value they take while respecting the constraints has the same cost. Thus, we can solve the complete formulation as follows: first, we solve the "partial" formulation in (41) and then pick $w_{ie} \in \{0,1\}$ to satisfy $\mu_{ie} = w_{ie}\delta_e$. It remains to be shown that we can indeed find such w_{ie}. The equation $\mu_{ie} = w_{ie}\delta_e$ (all variables are binary valued) has a solution for w_{ie} if $\mu_{ie} \le \delta_e$. The first set of constraints already ensures that this will be the case: since every term in the LHS of $\sum_{i\in\Omega_s^{\#}} \mu_{ie} = \delta_e$ is non-negative, each one of them is no larger than the RHS. Thus, any solution to (41) satisfies $\mu_{ie} \le \delta_e$, ensuring that the equations $\mu_{ie} = w_{ie}\delta_e$ can be solved.

REFERENCES

[1] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *Proc. ACM 2nd Int. Conf. Embedded Networked Sensor Syst.*, 2004, pp. 1–12.

[2] X. Wang, Z. Wang, and B. O'Dea, "A ToA-based location algorithm reducing the errors due to non-line-of-sight (NLOS) propagation," *IEEE Trans. Veh. Technol.*, vol. 52, no. 1, pp. 112–116, 2003.

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

[4] M. Gavish and A. Weiss, "Performance analysis of bearing-only target location algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 3, pp. 817–828, 1992.

[5] Y. Chan and K. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, 1994.

[6] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, 2008.

[7] X. Xu, N. Rao, and S. Sahni, "A computational geometry method for localization using differences of distances," *ACM Trans. Sensor Netw. (TOSN)*, vol. 6, no. 2, p. 10, 2010.

[8] A. Bishop, B. Fidan, K. Dogancay, B. Anderson, and P. Pathirana, "Exploiting geometry for improved hybrid AOA/TDOA-based localization," *Signal Process.*, vol. 88, no. 7, pp. 1775–1791, 2008.

[9] J. Chen, R. Hudson, and K. Yao, "Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1843–1854, 2002.

[10] Y. Shen and M. Win, "Fundamental limits of wideband localization Part I: A general framework," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4956–4980, 2010.

[11] Y. Shen, H. Wymeersch, and M. Win, "Fundamental limits of wideband localization part II: Cooperative networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4981–5000, 2010.

[12] M. Win, A. Conti, S. Mazuelas, Y. Shen, W. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, 2011.

[13] X. Li and K. Pahlavan, "Super-resolution ToA estimation with diversity for indoor geolocation," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 224–234, 2004.

[14] D. Dardari, M. Luise, and E. Falletti, *Satellite and Terrestrial Radio Positioning Techniques: A Signal Processing Perspective*. New York: Academic, 2011.

[15] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007.

[16] D. Dardari, "Pseudorandom active UWB reflectors for accurate ranging," *IEEE Commun. Lett.*, vol. 8, no. 10, pp. 608–610, 2004.

[17] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, 2005.
[18] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proc. IEEE*, vol. 97, no. 2, pp. 404–426, 2009.
[19] Y. Qi, H. Kobayashi, and H. Suda, "On time-of-arrival positioning in a multipath environment," *IEEE Trans. Veh. Technol.*, vol. 55, no. 5, pp. 1516–1526, 2006.
[20] J. Caffery, Jr. and G. Stuber, "Subscriber location in CDMA cellular networks," *IEEE Trans. Veh. Technol.*, vol. 47, no. 2, pp. 406–416, 1998.
[21] X. Xu, S. Sahni, and N. Rao, "On basic properties of localization using distance-difference measurements," in *Proc. IEEE 11th Int. Conf. Inf. Fusion*, 2008, pp. 1–8.
[22] D. Hubel and T. Wiesel, "Brain mechanisms of vision," *Sci. Amer.*, vol. 241, no. 3, p. 150, 1979.
[23] S. Venkateswaran and U. Madhow, "Space-time localization using times of arrival," in *Proc. Commun., Control, Comput. Conf. (Allerton)*, Sep. 2011, pp. 1544–1551.
[24] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," in *Proc. IEEE Comput. Soc. Int. Conf. Inf. Process. Sensor Netw.*, 2009, pp. 37–48.
[25] S. Sanghavi, "Equivalence of LP relaxation and max-product for weighted matching in general graphs," in *Proc. Inf. Theory Workshop*, 2007, pp. 242–247.
[26] S. Venkateswaran and U. Madhow, "Localizing multiple events using times of arrival: A parallelized, hierarchical approach to the association problem," Jun. 2012 [Online]. Available: www.ece.ucsb.edu/wcsl/Publications/multipleEventLocalizationTR.pdf
[27] J. Isaacs, D. Klein, and J. Hespanha, "Optimal sensor placement for time difference of arrival localization," in *Proc. IEEE 48th IEEE Conf. Decision Control*, 2009, pp. 7878–7884.
[28] D. Klein, S. Venkateswaran, J. T. Isaacs, J. Burman, T. Pham, J. Hespanha, and U. Madhow, "Localization with sparse acoustic sensor networks using UAVs as information seeking data mules," *ACM Trans. Sensor Netw.*, 2012, accepted for publication.
[29] A. Bishop, B. Fidan, B. Anderson, P. Pathirana, and K. Dogancay, "Optimality analysis of sensor-target geometries in passive localization: Part 2—Time-of-arrival based localization," in *Proc. 3rd Int. Conf. Intell. Sensors, Sensor Netw., Inf. (ISSNIP)*, Dec. 2007, pp. 13–18.
[30] J. Burman, J. Hespanha, U. Madhow, J. Isaacs, S. Venkateswaran, and T. Pham, "Autonomous UAV persistent surveillance using bio-inspired strategies," in *SPIE Defense, Secur., Sens. Symp.*, 2012, submitted for publication.
[31] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, "Target tracking with binary proximity sensors," *ACM Trans. Sensor Netw. (TOSN)*, vol. 5, no. 4, pp. 30–30, 2009.

Sriram Venkateswaran (M'12) received the Bachelor's degree in electrical engineering from the Indian Institute of Technology, Madras, in 2006 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara, in 2007 and 2011, respectively.

He is currently a Postdoctoral Researcher at the University of California, Santa Barbara. He interned at Qualcomm, Santa Clara, CA, during summer 2010. His research interests lie in applying estimation theory to problems in sensor and communication networks.

Upamanyu Madhow (F'05) received the Bachelor's degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1985 and the Ph.D. degree in electrical engineering from the University of Illinois, Urbana-Champaign, in 1990.

He formerly worked as a Research Scientist at Bell Communications Research, Morristown, NJ, and as a faculty at the University of Illinois, Urbana-Champaign. He is currently a Professor of Electrical and Computer Engineering at the University of California, Santa Barbara. He is the author of the textbook *Fundamentals of Digital Communication* (Cambridge Univ. Press, 2008). His research interests broadly span communications, signal processing, and networking, with current emphasis on millimeter wave communication and bio-inspired approaches to networking and inference.

Dr. Madhow is a recipient of the NSF CAREER award. He has served as Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON INFORMATION THEORY, and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.