

# Space-Time Localization using Times of Arrival

Sriram Venkateswaran and Upamanyu Madhow  
 Department of Electrical and Computer Engineering  
 University of California Santa Barbara, CA 93106, USA  
 Email: {sriram,madhow}@ece.ucsb.edu

**Abstract**—We provide a computationally efficient framework for utilizing Time of Arrival (ToA) sensors to localize *multiple* events in close proximity in space and time. Conventional ToA-based localization algorithms are typically designed for single events. Naïve approaches for associating ToAs with events and then applying a conventional localization algorithm incur complexity exponential in the number of sensors. An alternative approach of hypothesis testing over a space-time grid also has excessive complexity for large deployment regions. We propose an approach that sidesteps such computational bottlenecks by using discretization in time to efficiently generate a list of candidate events (including true as well as “phantom” events), and then employ statistical techniques to refine these estimates and to solve the ToA-to-event matching problem using linear programming on a bipartite graph. The algorithm automatically rejects phantom events and accounts for misses and outliers, providing performance close to that of a genie-based algorithm with ideal knowledge of ToA-to-event matching.

## I. INTRODUCTION

We consider the fundamental problem of space-time localization using Time-of-Arrival (ToA) sensors: given a list of ToAs at each sensor, can we estimate the location and time of occurrence of multiple events that can be closely spaced in time? Most existing fusion algorithms consider one event at a time, implicitly assuming that, if multiple events do occur, then they are well enough separated in time that there is no ambiguity in matching ToAs to events. However, ambiguities in matching are inevitable when events can occur in quick succession at different locations, and when there are outliers or misses in the sensor observations. In this paper, we provide a computationally efficient approach to localize events in space and time, resolving such ambiguities while avoiding combinatorial explosion in assigning ToAs to events.

As a concrete illustration, Figure 1 shows three events sensed by eight sensors where the ordering of the ToAs for two different events can be different at different sensors (this can easily happen, for example, with acoustic sensors because of the relatively low speed of sound). Thus, a naïve strategy of sorting ToAs at each sensor in order to assign them to events does not work. Alternatively, a brute force strategy is to try all possible groupings, run a standard ToA-based localization strategy for each grouping, and prune down to the “best” grouping. This is computationally unattractive: for  $E$  events detected at  $N$  sensors, there are  $E^N$  groupings. In addition, both of these strategies run into trouble when some of the sensors “miss” events of interest or pick up “outliers” due to spurious readings, in which case the number of ToA readings may vary across sensors. This forces us to implicitly

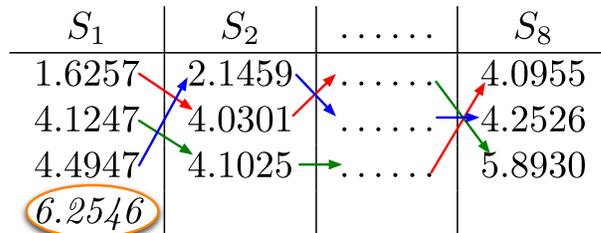


Fig. 1. Three events, that we call “Blue (B)”, “Green (G)” and “Red (R)”, happen close to one another in time and produce ToAs at 8 sensors. The ToAs at each sensor are sorted in ascending order. The red arrows connect the ToAs produced by the red event and so on. Note that the events need not arrive 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 to group the ToAs appropriately – we design an algorithm to draw the arrows that connect ToAs produced by the same event – and localize the events. 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.

estimate the number of events that occurred, in addition to their locations and times.

One way of avoiding the grouping problem is to discretize space and time into a grid, as in [1], and to evaluate the evidence provided by the ToA readings at all sensors for each grid point. While we comment on the specific algorithm in [1] in more detail shortly, the biggest drawback of this approach is that its computational complexity scales with the size of the deployment region. However, the idea of parallelizing evaluation of the evidence provided by the ToA readings, rather than first matching them with events, is also key to our strategy.

**Approach:** The critical step is to discretize the times at which events could have occurred. For a hypothesized time of occurrence  $t_e$ , a ToA reading  $t_s > t_e$  at a given sensor corresponds to the event lying on a circle of radius  $c(t_s - t_e)$  around the sensor, where  $c$  is the speed of propagation of the phenomenon of interest (e.g., the speed of sound for acoustic sensors, or the speed of light for radio frequency sensors). The intersection of circles produced by different ToAs at different sensors can be computed in closed form, and doing this for pairs of sensors enables us to rapidly generate a list of candidate events, some of which correspond to true events, while others are “phantom” events. While we are primarily motivated by acoustic sensing applications, the inspiration for our approach was actually provided by the center-surround neural responses characteristic of mammalian vision [2]. The intersection of such responses, together with further process-

ing, including feedback from higher layers, is what enables us to perceive complex scenes, and we proceed analogously for our problem as well. Once we have generated candidate events, we refine their locations by using a statistical goodness metric for merging “duplicates” and for discarding some of the phantoms. This leads to a *palette* containing refined versions of true and phantom event estimates, and this is when we address the problem that we have hitherto postponed, of grouping ToAs according to events by formulating it as a modified version of the matching problem on a bipartite graph. The events picked from the palette must be “matched” against the observations at different sensors, with the price of a matching provided by an approximation to the Maximum Likelihood (ML) cost function. We show that this problem can be posed as a binary integer program, which we then relax to solve a linear program and choose the events that actually occurred.

**Related Work:** Source localization is a classical problem with a rich body of literature. For example, algorithms using ToAs [3], AoAs [4], Time Differences of Arrival (TDoAs) [5], [6], hybrid versions of these (hybrid TDoA-AoA) [7] and wideband processing of recorded signals [8] have been proposed to localize sources. A more detailed set of references for each of these techniques can be found in the survey paper [9]. Most of this prior work does not apply to our setting, since it considers one event at a time. A notable exception is the algorithm for localizing multiple events using ToA sensors in [1], which discretizes the space-time grid, and counts how many sensors “agree” with a given grid point. Denoting the count corresponding to grid point  $(x, y, t)$  by  $C(x, y, t)$ , the event locations and times are estimated as local maxima of  $C(x, y, t)$ . The complexity grows with the size of the deployment region as well as the desired granularity of the space-time estimates, unlike our approach (which only discretizes time, and refines coarse estimates from the first stage using statistical criteria). Further, the work in [1] does not take explicit account of the problem constraints (every ToA must either be associated with a single event or declared to be an outlier), unlike the linear program based matching in our algorithm.

## II. SYSTEM MODEL

We wish to monitor a two-dimensional region  $\mathcal{D}$  and deploy  $N$  sensors inside  $\mathcal{D}$  at  $\theta_1, \theta_2, \dots, \theta_N (\theta_i \in \mathbb{R}^2)$ . Suppose that we observe the system over the time window  $[0, T]$  and  $E$  events occur in this period. We describe the  $e$ th among these  $E$  events by the triplet  $(\varphi_e, t_e)$  where  $\varphi_e \in \mathbb{R}^2$  denotes the spatial location of the event and the  $t_e$  denotes the time at which the event occurred. An event at  $(\varphi_e, t_e)$  is “heard” at sensor  $s$  with probability  $1 - p_{miss}$ , thereby producing a ToA, or is missed with probability  $p_{miss}$  and does not have any ToA corresponding to it. If an event at  $(\varphi_e, t_e)$  is heard at sensor  $s$ , it produces a ToA  $\tau(e \leftrightarrow s)$  given by,

$$\tau(e \leftrightarrow s) = t_e + \frac{\|\varphi_e - \theta_s\|}{c} + n \quad (1)$$

where  $c$  denotes the speed of sound,  $\|\cdot\|$  denotes the two-norm of a vector and  $n$  is the measurement noise, assumed to be distributed as  $N(0, \sigma^2)$ . Furthermore, we assume that the

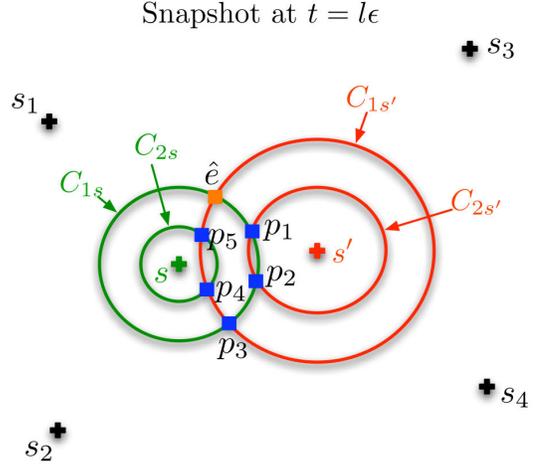


Fig. 2. Geometry of the processing in Stage 1. Six sensors  $s, s', s_1, s_2, s_3, s_4$  are shown. Sensors  $s$  and  $s'$  have two ToAs each, denoted by  $\{\tau_1(s), \tau_2(s)\}$  and  $\{\tau_1(s'), \tau_2(s')\}$ . ToAs  $\tau_1(s)$  and  $\tau_1(s')$  were produced by an event  $\mathcal{E}$  that occurred at time  $t_e \approx l\epsilon$ . Consider a hypothesized event time  $u = l\epsilon$  and draw circles  $C_{1s}$  and  $C_{2s}$ , centered at sensor  $s$ , with radii  $\tau_s(1) - u$  and  $\tau_s(2) - u$  (likewise for  $C_{1s'}$  and  $C_{2s'}$ ).  $C_{1s}$  and  $C_{1s'}$  intersect at a point  $\hat{e}$  close to  $\mathcal{E}$ 's location. All other points of intersection between  $C_{is}$  and  $C_{js'} \forall i, j$  (denoted by  $p_i, i = 1, \dots, 5$ ) are phantom estimates.

measurement noises corresponding to different event-sensor pairs are independent. We set  $c = 1$  without loss of generality, based on a suitable redefinition of the units for distance and time. With this change, the measurement model in (1) simplifies to,

$$\tau(e \leftrightarrow s) = t_e + \|\varphi_e - \theta_s\| + n. \quad (2)$$

For our statistical processing, we model event occurrence as a space-time Poisson process, with events occurring at rate  $\lambda_{LS}$  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 therefore do not model the locations of such events and model their ToAs as arising from a Poisson process with a rate  $\lambda_O$  (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  $i$ 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  $\Omega_s = \{\tau_s(1), \tau_s(2), \dots, \tau_s(M_s)\}$ , with  $\tau_s(i) \leq \tau_s(j)$  whenever  $i \leq j$ . Note that the number of ToAs can vary across sensors because misses and outliers occur independently at each sensor.

## III. ALGORITHM : THE BIG PICTURE

In this section, we provide a bird’s-eye view of the proposed algorithm, which can be divided into three stages.

• **Stage 1:** The goal of the first stage is to quickly generate a number of candidate events that are “reasonably good”. To

do this, we first discretize the times at which large-scale events might have occurred and suppose that they only take the values  $(\dots, -2\epsilon, -\epsilon, 0, \epsilon, 2\epsilon, \dots)$ . Next, we hypothesize that an event at  $t = l\epsilon$  produced the  $i$ th ToA at sensor  $s$  and  $j$ th ToA at sensor  $s'$ . Under these hypotheses, it is easy to show that the event must be located at the intersection of two circles – the circles are centered at the sensor locations  $\boldsymbol{\theta}_s$  and  $\boldsymbol{\theta}_{s'}$  and their radii are given by  $\tau_s(i) - l\epsilon$  and  $\tau_{s'}(j) - l\epsilon$  respectively. Since the intersection of two circles can be specified in closed-form, we can generate the event location easily. By repeating this process for other values of  $(l, s, i, s', j)$ , we generate a number of candidate event locations.

What does this list of candidates look like? Suppose that an event  $(\varphi_e, t_e)$  produced the ToAs  $\tau_s(i)$  and  $\tau_{s'}(j)$  at sensors  $s$  and  $s'$  in reality. Consider a hypothesized event at a time  $u$  that is close to  $t_e$ , say  $u = \epsilon \lfloor t_e/\epsilon \rfloor$ . The event location produced by intersecting circles centered at  $\boldsymbol{\theta}_s$  and  $\boldsymbol{\theta}_{s'}$  with radii  $\tau_s(i) - u$  and  $\tau_{s'}(j) - u$  will be close to the true event location  $\varphi_e$ . Thus, estimates close to true event locations will be a part of the list of candidates. On the other hand, suppose that one of hypotheses is false – either the ToAs  $\tau_s(i)$  and  $\tau_{s'}(j)$  are *not* produced by the same large scale event or even if they are, the event is not at the hypothesized time  $l\epsilon$ . In such cases, intersecting circles centered at  $\boldsymbol{\theta}_s$  and  $\boldsymbol{\theta}_{s'}$  with radii  $\tau_s(i) - l\epsilon$  and  $\tau_{s'}(j) - l\epsilon$  produces an estimated event location; but, such an event clearly did not occur in reality. We call such estimates “phantoms”. Figure 2 shows an example of true and phantom estimates being generated by intersecting circles drawn at sensors  $s$  and  $s'$ .

To discard phantom estimates from the list, we compute a “goodness” metric for each of these events. This metric uses the measurements at all the sensors to capture the likelihood of the event having happened. We discard candidate estimates whose goodness falls below a threshold from the list. Since many phantoms have a low value of goodness, the list is pruned considerably. However, we choose the threshold conservatively to ensure that no event estimate close to a true event is discarded (in spite of non-idealities such as noise, misses at some sensors etc.). As a result, some phantom estimates survive the pruning process and remain on the list of candidate events. Therefore, the output of Stage 1 is a list of candidate events, containing both true and phantom estimates.

• **Stage 2:** An event estimate from Stage 1, produced by intersecting circles centered at sensors  $s$  and  $s'$ , is bound to be noisy since it does not take measurements at other sensors into account. In the second stage, we use measurements at all the sensors to refine this noisy estimate. We do this in an iterative fashion. In the first iteration of Stage 2, we use the event estimates from Stage 1 as a starting point and linearize the constraints imposed by the ToA measurements about this point at all sensors. This allows us to refine the estimate with low-complexity. In subsequent iterations of Stage 2, we use the estimates from the previous iteration as the starting point to further refine them. The output of this stage, therefore, is a *palette* of events, that contains refined versions of both true and phantom event estimates.

• **Stage 3:** The goal of Stage 3 is to pick the true event estimates from the overcomplete palette of candidate events, containing both true and phantom event estimates. To do this, we pick the *subset* of events from the palette that fit the observations at all the sensors in the “best” possible fashion. In principle, this can be done by brute-force: pick a subset of events from the palette, hypothesize that this subset contains all the true events that occurred and no phantom events, evaluate the likelihood of the observations under this hypothesis and pick the subset with the largest likelihood as the estimate of events that occurred. However, we show that it can be solved far more efficiently by posing it as a problem on a bipartite graph. The events in the palette form the first set of nodes in the graph and the observations at different sensors form the other set. The objective is to pick events from the palette and then add edges to the graph, pairing the “picked events” with the observations at different sensors. Each edge comes with a cost, which captures the likelihood that the event at one end of the edge generated the observation at the other end. Furthermore, we need to satisfy some constraints while adding the edges. Specifically, the two sets of constraints are: (i) at each sensor, an event must either produce a ToA or declared to be missed and (ii) every ToA must either be associated with an event or declared to be an outlier. We capture such constraints in a binary integer programming problem. Finally, we relax the integer program and solve a linear problem to pick the “most likely” subset of events from the palette.

We now provide the details for each of these stages.

#### IV. STAGE 1: GENERATING CANDIDATE EVENTS

Consider the case when the measurements are noiseless and suppose that  $\tau_s(i)$ , the  $i$ th ToA at sensor  $s$ , and  $\tau_{s'}(j)$ , the  $j$ th ToA at sensor  $s'$ , are produced by the same event that occurred at time  $u = l\epsilon$ . From (2), the location of the hypothesized event, denoted by  $\mathbf{r}$ , must satisfy

$$u + \|\mathbf{r} - \boldsymbol{\theta}_s\| = \tau_s(i) \Rightarrow \|\mathbf{r} - \boldsymbol{\theta}_s\| = \tau_s(i) - u \quad (3)$$

$$u + \|\mathbf{r} - \boldsymbol{\theta}_{s'}\| = \tau_{s'}(j) \Rightarrow \|\mathbf{r} - \boldsymbol{\theta}_{s'}\| = \tau_{s'}(j) - u \quad (4)$$

From these equations, we see that 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. Denoting the points of intersection by  $\mathbf{r}_+$  and  $\mathbf{r}_-$ , we have,

$$\mathbf{r}_\pm = \frac{\boldsymbol{\theta}_s + \boldsymbol{\theta}_{s'}}{2} - \frac{R_{s'}^2 - R_s^2}{2d_{ss'}^2}(\boldsymbol{\theta}_{s'} - \boldsymbol{\theta}_s) \pm \frac{b}{2d_{ss'}^2}(\boldsymbol{\theta}_{s'}^\perp - \boldsymbol{\theta}_s^\perp) \quad (5)$$

where  $R_s$  and  $R_{s'}$  are the radii of the circles  $\tau_s(i) - u$  and  $\tau_{s'}(j) - u$ ,  $d_{ss'}$  is the distance between the sensors  $\|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s'}\|$ ,  $b = \sqrt{[(R_s + R_{s'})^2 - d_{ss'}^2][d_{ss'}^2 - (R_s - R_{s'})^2]}$  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  $\boldsymbol{\theta}_{s'} - \boldsymbol{\theta}_s$ .

If an event at  $(\varphi_e, t_e)$  indeed produced *both* the ToAs  $\tau_s(i)$  and  $\tau_{s'}(j)$ , and the hypothesized event time  $u = l\epsilon \approx t_e$ , then one of the location estimates  $\mathbf{r}_+$  or  $\mathbf{r}_-$  will be close to  $\varphi_e$  and the other will be a phantom estimate. On the other hand,

if  $\tau_s(i)$  and  $\tau_{s'}(j)$  are not produced by the same event or the hypothesized event time is not close to that of any event, both  $(\mathbf{r}_+, u)$  and  $(\mathbf{r}_-, u)$  are phantom estimates that we should discard ultimately.

**Accuracy of Estimates:** We now calculate the covariance of the spatial location estimates  $\mathbf{r}_\pm$  when the ToA measurements are noisy and use it to compute the “goodness” of the estimated events  $(\mathbf{r}_\pm, u)$ . Suppose that the ToAs  $\tau_s(i)$  and  $\tau_{s'}(j)$  are corrupted by measurement noises  $n_s(i)$  and  $n_{s'}(j)$  respectively, leading to corresponding errors  $\mathbf{e}_\pm$  in the spatial event estimates (5). Assuming that the measurement noises are small and using a Taylor series expansion of (5), we get,

$$\mathbf{e}_\pm = \mathbf{K}_\pm \begin{bmatrix} n_s(i) \\ n_{s'}(j) \end{bmatrix} \text{ where,}$$

$$\mathbf{K}_\pm = \mathbf{M} \begin{bmatrix} \frac{R_s}{d_{ss'}} & \frac{-R_{s'}}{d_{ss'}} \\ \frac{R_s(d_{ss'} - \beta)}{d_{ss'} \sqrt{R_s^2 - \beta^2}} q_\pm & \frac{\beta R_{s'}}{d_{ss'} \sqrt{R_s^2 - \beta^2}} q_\pm \end{bmatrix} \quad (6)$$

with  $\mathbf{M} = \begin{bmatrix} \frac{\theta_{s'} - \theta_s}{d_{ss'}} & \frac{\theta_{s'}^\perp - \theta_s^\perp}{d_{ss'}} \end{bmatrix}$ ,  $\beta = (R_s^2 - R_{s'}^2 + d_{ss'}^2)/2d_{ss'}$  and  $q_\pm = \text{sign}((\mathbf{r}_\pm - \theta_s)^T(\theta_{s'}^\perp - \theta_s^\perp))$ . Using the fact that  $n_s(i)$  and  $n_{s'}(j)$  are independent Gaussian random variables, we obtain the covariance matrices of the estimates to be  $\mathbf{C}_\pm = \mathbb{E}(\mathbf{e}_\pm \mathbf{e}_\pm^T) = \sigma^2 \mathbf{K}_\pm \mathbf{K}_\pm^T$ .

We repeat this process for other choices of ToA pairs and hypothesized event times, resulting in a list of candidate events  $\{(\mathbf{r}_n, u_n)\}$  with covariance matrices  $\{\mathbf{C}_n = \sigma^2 \mathbf{K}_n \mathbf{K}_n^T\}$ ,  $n = 1, 2, \dots$ . As explained before, some of these estimates are close to events that occurred in reality, while others are simply phantom estimates. We now compute a goodness metric for each of these candidates to quickly discard those that are “obviously” phantom estimates from the list.

**Goodness Metric:** Consider a candidate event  $\mathcal{E} = (\mathbf{r}, u)$  with a covariance matrix  $\mathbf{C} = \sigma^2 \mathbf{K} \mathbf{K}^T$ . The goodness metric for  $\mathcal{E}$  is designed to capture the likelihood of the observations at all the sensors, assuming that  $\mathcal{E}$  happened. We compute the goodness in two steps: first, we calculate individual goodnesses for  $\mathcal{E}$  at each sensor  $s$  and then, multiply them out to obtain an overall goodness. The basic steps in computing the goodness for the event  $\mathcal{E}$  at a generic sensor  $s$  are as follows:

1. Assuming that  $\mathcal{E}$  happened, we predict the expected ToA that sensor  $s$  must have seen, given by  $\eta_s = u + \|\mathbf{r} - \theta_s\|$ .

2. We compare the ToAs observed at sensor  $s$ , given by  $\Omega_s = \{\tau_s(1), \dots, \tau_s(M_s)\}$ , with the predicted ToA  $\eta_s$  and pick the one that is closest to  $\eta_s$ . We denote this ToA by  $\tau_s(\mathcal{E})$ , i.e.,

$$\tau_s(\mathcal{E}) = \arg \min_i |\tau_s(i) - \eta_s| \quad (7)$$

Loosely speaking,  $\tau_s(\mathcal{E})$  is the best evidence that the sensor  $s$  has to offer for the event  $\mathcal{E}$  having taken place.

3. The goodness at sensor  $s$  depends solely on the difference between the predicted ToA  $\eta_s$  and the observed ToA  $\tau_s(\mathcal{E})$ , given by  $z_s = \tau_s(\mathcal{E}) - \eta_s$ . We denote the goodness for  $\mathcal{E}$  at sensor  $s$  by  $L(z_s)$ . Intuitively, we expect  $L(z_s)$  to be large if the mismatch  $z_s$  is small and decrease monotonically

as  $z_s$  increases.

We compute  $L(z_s)$  in two steps: first, we condition 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 the probability of miss, as

$$L(z_s) = (1 - p_{miss})L(z_s|\mathcal{E} \text{ heard}) + p_{miss}L(z_s|\mathcal{E} \text{ missed}) \quad (8)$$

Owing to a lack of space, we only provide a sketch of the steps used to compute  $L(z_s|\mathcal{E} \text{ heard})$  and  $L(z_s|\mathcal{E} \text{ missed})$ .

Suppose that the event  $\mathcal{E} = (\mathbf{r}, u)$  is heard at sensor  $s$ . Then, the difference  $z_s$  between the predicted ToA  $u + \|\mathbf{r} - \theta_s\|$  and the best evidence  $\tau_s(\mathcal{E})$  has two sources: (1) The location estimate  $\mathbf{r}$  from Stage 1 has an error  $\mathbf{e}$  with covariance  $\mathbf{C}$ . This leads to an error  $\Delta$  in the predicted ToA  $u + \|\mathbf{r} - \theta_s\|$ . (2) The measurement  $\tau_s(\mathcal{E})$  is corrupted by Gaussian noise  $n \sim N(0, \sigma^2)$ . Under the assumption that measurement noise is small, we can show that (a)  $z_s = \Delta + n$  has a zero-mean Gaussian distribution and (b) the variance of  $\Delta$ , denoted by  $\sigma_\Delta^2$  is equal to  $\sigma^2 \frac{\|(\mathbf{r} - \theta_s)^T \mathbf{K}\|^2}{\|\mathbf{r} - \theta_s\|^2}$  where  $\sigma \mathbf{K}$  is the “square-root” of the covariance  $\mathbf{C}$  of the spatial estimate  $\mathbf{r}$  from Stage 1, so that  $\mathbf{C} = \sigma^2 \mathbf{K} \mathbf{K}^T$ . Neglecting any correlation between  $\Delta$  and  $n$ , the variance of  $z_s$  is given by  $\sigma_s^2 = \sigma_\Delta^2 + \sigma^2$ . Therefore, we have

$$L(z_s|\mathcal{E} \text{ heard}) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{z_s^2}{2\sigma_s^2}\right) \quad (9)$$

Next, consider the computation of  $L(z_s|\mathcal{E} \text{ missed})$ . When the event  $\mathcal{E}$  is missed at sensor  $s$ , the observation  $z_s$  is obtained by taking the difference between the predicted ToA for  $\mathcal{E}$ , denoted by  $\eta_s$ , and an observed ToA  $\tau_s(\mathcal{E})$  produced by a *completely different event*. Furthermore, we note that  $\tau_s(\mathcal{E})$  is the closest among all ToAs recorded at sensor  $s$  to  $\eta_s$  (see (7)). Two conditions must be satisfied for this to have happened: (a) there must be no ToAs at sensor  $s$  that are closer to  $\eta_s$  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  $[\eta_s + z_s, \eta_s + z_s + dz]$ . We note that each sensor observes ToAs being generated at a rate  $\lambda = \lambda_{LS}(1 - p_{miss}) + \lambda_O$ . We make the further assumption that these ToAs arrive according to a Poisson process with rate  $\lambda$ . With this assumption, the probability of (a) happening is  $\exp(-2\lambda|z_s|)$  and the likelihood of (b) is  $\lambda dz$ . Since the time intervals considered in (a) and (b) do not overlap, the events are independent and we obtain,

$$L(z_s|\mathcal{E} \text{ missed})dz = \exp(-2\lambda|z_s|)\lambda dz \quad (10)$$

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

Putting (8),(9) and (10) together, we get the goodness at sensor  $s$  to be,

$$L(z_s) = \frac{(1 - p_{miss})}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{z_s^2}{2\sigma_s^2}\right) + p_{miss}\lambda \exp(-2\lambda|z_s|) \quad (11)$$

Note that 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  $\exp(-|z_s|^2)$ . Therefore, the net effect of this term is to ensure that  $L(z_s)$  does not become too small even when  $z_s$  is fairly large. Furthermore, typical values of  $1/\lambda$  (for example, a value of  $5/3$  corresponding to 3 events in 5 seconds) are much larger than  $\sigma$  (typically 0.01 s), further slowing down the decay of the second term relative to the first. Therefore, for all practical purposes, we can neglect the decay in the second term and simply approximate  $L(z_s)$  as,

$$L(z_s) \approx \frac{(1 - p_{miss})}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{z_s^2}{2\sigma_s^2}\right) + p_{miss}\lambda \quad (12)$$

The overall goodness for the event  $\mathcal{E}$ , denoted by  $g$ , is

$$g = \sum_{s=1}^N \log(L(z_s)) \quad (13)$$

If the goodness  $g$  falls below a threshold  $\kappa$ , we declare the event to be a phantom and discard it from the candidate list.

**Choosing the threshold  $\kappa$ :** If sensor  $s$  “hears” the event  $\mathcal{E}$ , so that  $z_s$  is on the order of  $\sigma_s$ , the first term in the expression for  $L(z_s)$  dominates the second. On the other hand, if sensor  $s$  misses the event,  $z_s$  is typically large and the first term dies to zero fairly quickly (say,  $|z_s| \geq 4\sigma_s$ ). In such cases, we can approximate  $L(z_s)$  by  $p_{miss}\lambda$ . We choose the threshold  $\kappa$  so that a candidate event survives the pruning process only if it is heard at  $N_{hear}$  sensors (at least). Guided by the above observations, we set the threshold  $\kappa$  to be

$$\kappa = N_{hear} \left[ \log\left(\frac{1}{2\pi\sigma^2}\right) - \frac{1.5^2}{2} \right] + (N - N_{hear})(p_{miss}\lambda). \quad (14)$$

Note that we have used the fact that  $|z_s|$  is on the order of  $1.5\sigma_s$  when a sensor hears an event to set the threshold.

**Clustering via ToA groupings:** While we have discarded events that are obviously phantoms by thresholding the goodness, the list can be pruned further by eliminating events that are essentially “poorer duplicates” of other events on the list. To understand the origins of duplication in the list of candidate events, consider the following example. 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  $\mathbf{r}_a$ . Consider the “next” hypothesized event time  $u' = u + \epsilon$ . If  $\epsilon$  is “small”, intersecting circles of radii  $\tau_s(i) - u'$  and  $\tau_{s'}(j) - u'$  centered at sensors  $s$  and  $s'$  will result in an event estimate  $\mathbf{r}_b$  that is very close to  $\mathbf{r}_a$ . It is clear that  $(\mathbf{r}_a, u)$  and  $(\mathbf{r}_b, u')$  are estimates of the same event and it is sufficient to store the “better” estimate among the two. We could do this by using a standard clustering algorithm which groups estimates that are close in space and time and stores only the “best” representative from each group. However, we exploit the structure of the problem to cluster these events in a principled fashion by introducing the concept of a *grouping*.

Consider an event  $\mathcal{E} = (\mathbf{r}, u)$  whose goodness is above the threshold. As before, we 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 observed ToA at sensor  $s$  that is closest to  $\eta_s$ ). The grouping  $\mathbf{p}$  associated with an event  $\mathcal{E}$  is a set of  $N$  quantities  $\{p_1, p_2, \dots, p_N\}$  where  $p_s$  stores the evidence  $\tau_s(\mathcal{E})$  if it is “compelling”; otherwise, it records the fact that sensor  $s$  has missed the event. Specifically, if the difference  $|\tau_s(\mathcal{E}) - \eta_s|$  is smaller than a threshold  $\gamma$ , we set  $p_s = \tau_s(\mathcal{E})$ ; otherwise, we store the string *miss* in  $p_s$ . Duplicated events, such as  $\mathbf{r}_a$  and  $\mathbf{r}_b$  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 (13)).

## V. 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 and we do not account for the measurements at other sensors, the estimates can have significant errors. In this section, we use the measurements at all sensors to refine such noisy estimates.

Let  $\mathcal{E} = (\mathbf{r}, u)$  be a generic 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, letting  $\eta_s = u + \|\mathbf{r} - \boldsymbol{\theta}_s\|$  denote the predicted ToA for  $\mathcal{E}$  at sensor  $s$  and  $\tau_s(\mathcal{E})$  be the corresponding “best fit” ToA, we use sensor  $s$  in the refinement process only if  $|\tau_s(\mathcal{E}) - \eta_s| \leq \gamma$ .

**Refinement Procedure:** Suppose that the sensors  $s_1, s_2, \dots, s_Q$  have ToAs that are within  $\gamma$  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  $\tau_{s_1}(\mathcal{E}), \tau_{s_2}(\mathcal{E}), \dots, \tau_{s_Q}(\mathcal{E})$  respectively. Since the refined estimate, denoted by  $(\mathbf{r} + \Delta\mathbf{r}, u + \Delta u)$ , must fit the measurement model, we have,

$$\tau_{s_j}(\mathcal{E}) = u + \Delta u + \|\mathbf{r} + \Delta\mathbf{r} - \boldsymbol{\theta}_{s_j}\| + n_{s_j} \quad (15)$$

where  $n_{s_j} \sim N(0, \sigma^2)$  and  $j = 1, 2, \dots, Q$ . 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  $\Delta\mathbf{r}$  and retain only the linear term to approximate it as,

$$\|\mathbf{r} + \Delta\mathbf{r} - \boldsymbol{\theta}_{s_j}\| \approx \|\mathbf{r} - \boldsymbol{\theta}_{s_j}\| + \left\langle \Delta\mathbf{r}, \frac{\mathbf{r} - \boldsymbol{\theta}_{s_j}}{\|\mathbf{r} - \boldsymbol{\theta}_{s_j}\|} \right\rangle \quad (16)$$

where  $\langle \cdot, \cdot \rangle$  denotes the standard inner product. With this approximation, (15) can be rewritten as,

$$\tau_{s_j}(\mathcal{E}) - u - \|\mathbf{r} - \boldsymbol{\theta}_{s_j}\| = \left[ \frac{\mathbf{r}^T - \boldsymbol{\theta}_{s_j}^T}{\|\mathbf{r} - \boldsymbol{\theta}_{s_j}\|} \quad 1 \right] \begin{bmatrix} \Delta\mathbf{r} \\ \Delta u \end{bmatrix} + n_{s_j} \quad (17)$$

for  $j = 1, 2, \dots, Q$ . Let  $\mathbf{y}$  denote the  $Q$  dimensional vector whose  $j$ th component is  $\tau_{s_j}(\mathcal{E}) - u - \|\mathbf{r} - \boldsymbol{\theta}_{s_j}\|$  and  $H$  denote

the  $Q \times 3$  matrix whose  $j$ th row is  $\begin{bmatrix} \frac{\mathbf{r}^T - \theta_{s_j}^T}{\|\mathbf{r} - \theta_{s_j}\|} & 1 \end{bmatrix}$ . Then, the least-squares estimate of  $[\Delta \mathbf{r} \ \Delta u]$  is given by,

$$\begin{bmatrix} \Delta \hat{\mathbf{r}} \\ \Delta \hat{u} \end{bmatrix} = (H^T H)^{-1} H^T \mathbf{y} \quad (18)$$

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  $\gamma$  – 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.

To conclude, the output of stage 2 is a palette of candidate events that are refined versions of the estimates from stage 1. Note that the palette contains both true and phantom event estimates.

### VI. STAGE 3: PICKING TRUE EVENTS FROM THE PALETTE

In this stage, we pose and solve a problem on a bipartite graph to discard phantom events from the overcomplete palette and only retain the ones that occurred. The events in the palette, denoted by  $\mathcal{E}_1 = (\mathbf{r}_1, u_1), \dots, \mathcal{E}_P = (\mathbf{r}_P, u_P)$ , form one set of nodes of the bipartite graph and the observations at the sensors form the other set of nodes. Figure 3 shows an example of such a graph. We represent the events in the palette by the blue circles in the uppermost row of the graph. Each observed ToA is denoted by a blue star and the ToAs observed at a sensor are arranged in a column. In the example shown in Figure 3, there are 4 sensors and they have  $\{3, 2, 4, 3\}$  observations respectively. The nodes marked  $M_1, M_2, M_3, M_4$  are “miss” nodes, that serve as proxies for any observations that might have been missed at the sensors. Similarly, the node marked Outlier acts as a representative for small-scale events that generate outlier observations at different sensors. The goal of Stage 3 is to pick a subset of the events  $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_P$  from the palette and associate them with the observations at the different sensors, thereby establishing a correspondence between the two “halves” of the bipartite graph. The problem of picking the “most reasonable” correspondence can be phrased as a constrained binary integer program and we provide the details now.

The first set of decisions we must make are: for each  $e = 1, 2, \dots, P$ , did event  $\mathcal{E}_e$  happen or is it a phantom? We refer to the events that we declared to have happened ( $\delta_e = 1$ ) as “picked events”. The second row of the graph in Figure 3 shows 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  $\mathcal{E}_5$ ).

The next task is to establish a correspondence between the picked events and the observations at various sensors. First, we identify the constraints that a “valid” correspondence must satisfy. Consider a picked event  $\mathcal{E}_e$  (such as  $\mathcal{E}_3$  in Fig. 3).

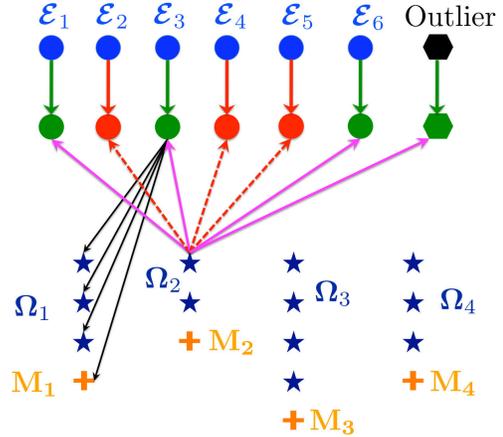


Fig. 3. Modified version of matching problem on a bipartite graph. Events in the palette are shown as blue circles and the observations at sensors are shown as blue stars. Green circles represent events that are picked while red circles denote phantom events. We need to draw edges between the picked events and the observations, subject to constraints, so as to maximize the sum of the values of the edges.

For each sensor  $s$ , we must decide on two things with regard to  $\mathcal{E}_e$ : (a) Was  $\mathcal{E}_e$  heard/missed at sensor  $s$  and (b) if it was heard, which observation did it generate? These decisions can be neatly summarized in the graph of Figure 3. Suppose that we draw edges between the event  $\mathcal{E}_e$  and all the observation nodes at sensor  $s$  (this includes blue stars representing the ToAs in  $\Omega_s$  and the miss node  $M_s$ , denoted by an orange cross). The four black lines in Figure 3 connecting  $\mathcal{E}_3$  to  $\tau_1(1), \tau_1(2), \tau_1(3)$  and  $M_1$  are examples of such edges. We can now provide the answers to (a) and (b) by “activating” exactly one of these edges and specifying which one it is; for example, if decide that  $\mathcal{E}_e$  was missed at sensor  $s$ , we activate the edge joining  $\mathcal{E}_e$  and  $M_s$  ( $\mathcal{E}_3$  and  $M_1$  in the example of Fig. 3). On the other hand, if we decide that  $\mathcal{E}_e$  was heard at sensor  $s$ , we activate the edge between  $\mathcal{E}_e$  and the appropriate observation in  $\Omega_s$  (one of the edges between  $\mathcal{E}_3$  and  $\tau_1(1), \tau_1(2)$  and  $\tau_1(3)$  in this example). Note that we need to specify such a correspondence only for the picked events (green circles) and not for phantom estimates (red circles). Next, we consider the point of view of an observed ToA at one of the sensors (for example, the first observation at sensor 2, denoted by  $\tau_2(1)$ , in Figure 3). This ToA must either be generated by a picked event or it must be an outlier. As before, we draw edges between the observation node and all the events in the palette (shown by pink and red lines in Fig. 3). If we declare this ToA 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 a picked event, 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 omit a formal statement of these constraints owing to a lack of space.

**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 using the following fact: given the decisions (picking events and choosing the correspondence), the observations at two sensors  $s$  and  $s'$  are independent. This allows us to phrase the problem of picking the “most likely” decisions in a simple manner, using the graphical model of Figure 3: (1) We assign a value to each edge that connects an event node and an observation node. This value captures the likelihood – in fact, it is equal to the logarithm of the likelihood – that the observation was produced by the event. (2) 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.

Consider an edge between an event node  $\mathcal{E}_e$  and the miss node  $M_s$  at sensor  $s$ . The value of this edge is  $\log(p_{miss})$ , since the chance that an event is missed at any sensor is  $p_{miss}$ . Next, consider the edge between event  $\mathcal{E}_e = (\mathbf{r}_e, u_e)$  and the  $j$ th observation at sensor  $s$ ,  $\tau_s(j)$ . For  $\mathcal{E}_e$  to produce this observation, two things must happen – (1)  $\mathcal{E}_e$  must be heard at sensor  $s$  and (2) given that it was heard at sensor  $s$ , it must produce the observation  $\tau_s(j)$ . The chance that (1) happens is  $1 - p_{miss}$  and the likelihood of (2) happening is exactly equal to the probability that the measurement noise accounts for the difference between the observed ToA  $\tau_s(j)$  and the predicted ToA  $u_e + \|\mathbf{r}_e - \boldsymbol{\theta}_s\|$ . Putting them together, we get the value of the edge between  $\mathcal{E}_e$  and  $\tau_s(j)$  to be

$$\text{val}(\mathcal{E}_e, \tau_s(j)) = \left( \log \frac{1 - p_{miss}}{\sqrt{2\pi\sigma^2}} \right) - \frac{(\tau_s(j) - u_e - \|\mathbf{r}_e - \boldsymbol{\theta}_s\|)^2}{2\sigma^2} \quad (19)$$

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  $\lambda_O$ , the chance that there are  $k$  outliers at sensor  $s$  over an observation window of length  $T$  is  $e^{-\lambda_O T} (\lambda_O T)^k / k!$ . The logarithm of this quantity, denoted by  $L$ , is given by (ignoring constants),

$$L = k \log(\lambda_O T) - \log(k!) \quad (20)$$

To see the problem, let us pretend that the term  $\log(k!)$  is absent. Then, this equation has a very simple interpretation: declaring an observation to be an outlier has the value  $\lambda_O T$  and the overall value of declaring  $k$  observations at a sensor to be 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  $\text{Poisson}(\lambda_O T)$  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. Specifically, we choose  $q$  so that the mean-square error between the sequences  $\{(1 - q)q^n, n = 0, 1, \dots\}$

and  $\{e^{-\lambda_O T} (\lambda_O T)^n / n!, n = 0, 1, \dots\}$  is minimized.

We can pose the problem of maximizing the cost function subject to the constraints as a binary integer program (all variables either take the value 0 or 1). We omit a formal statement of the integer program owing to a lack of space. 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  $N = 8$  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 can shed light on the performance of our algorithm.) Finally, we declare the events “picked” by the LP to have taken place.

## VII. SIMULATION RESULTS

*Sensor Deployment and Event Generation Models:* We simulate a deployment of  $N = 8$  sensors, with their positions chosen randomly in a circular region of radius  $R = 1020$  meters. We generate  $E = 3$  large-scale events, with their times chosen 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 90%. We generate outliers at a rate  $\lambda_O = 3/100$  events/sec at each sensor.

*Measurement Model:* We set the speed of sound  $c$  to 340 m/s and the standard deviation of the noise in the measurements  $\sigma$  to 0.02 s. Sensors miss an event with a probability  $p_{miss} = 5\%$ .

*Algorithm Choices:* We set the granularity of the hypothesized event times at  $\epsilon = 0.04$ s. We assume that every sensor observes ToAs arriving at a rate  $\lambda = 63/100 (= 3/5 + 3/100)$  events/sec and use it to compute the goodness. We set the minimum number of sensors that need to hear an event for it to survive in the list of candidates (see Stage 1) to  $N_{hear} = 4$  and compute the threshold  $\kappa$  (see (14)) accordingly. Finally, we choose  $q$  – the parameter of the geometric distribution that approximates the  $\text{Poisson}(\lambda_O T)$  distribution – to be 0.14.

We run 100 trials of the algorithm with the above parameters. In all the trials, the algorithm correctly estimates the number of events to be 3. The localization error for all the events (3 events/iteration  $\times$  100 iterations) is shown by the dotted blue line in Figure 4. The average localization error is 5.6 meters with a minimum error of 0.44 meters and a maximum error of 18.8 meters. To benchmark the performance of our algorithm, we use the following “genie”. For each large-scale event  $(\varphi_e, t_e)$ , we pick the ToAs produced by this event at various sensors. We then form the *Time Difference of Arrivals* (TDoAs) by taking the difference between the ToAs observed at different sensors and the one seen at the “first” sensor. These TDoAs are solely a function of the event location and we use them to localize the source by a brute force search over “reasonably good” candidates. Specifically, we discretize

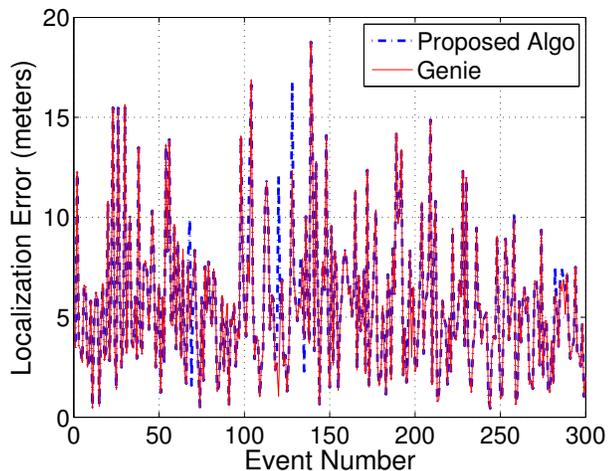


Fig. 4. Localization errors with the proposed algorithm and a genie-based scheme. The errors virtually coincide with one another, demonstrating the efficacy of the proposed algorithm.

a  $70\text{m} \times 70\text{m}$  region around the true source location  $\varphi_e$  into a  $2000 \times 2000$  grid of points. We choose the point in the grid that best fits the TDoAs at different sensors in the least-squares sense as the estimate of the source (this is also the ML estimate if the measurement noise is Gaussian). The solid red line in Figure 4 shows the localization errors observed with this genie based approach. We see that the errors produced by the proposed algorithm and the genie virtually coincide with one another, demonstrating the efficacy of the proposed scheme. Specifically, the mean localization error obtained with the genie based scheme is 5.50 m, only 0.1m lower than that obtained with the proposed algorithm.

**Impact of Multiple Events:** These results suggest that the algorithm is effective in grouping the ToAs that belong to an event, in the presence of “interference” from ToAs due to other events, and localizing it. However, on a few occasions, the grouping of ToAs across sensors is not “ideal”. We illustrate this with an example. In localizing  $\mathcal{E}_2$ , the second event, we use the ToAs it generated at all the sensors, except at sensor 4; at this sensor, we use the ToA produced by  $\mathcal{E}_3$ . Simultaneously, in localizing  $\mathcal{E}_3$ , we use the “correct” ToAs at all sensors but sensor 4, where we choose the ToA produced by  $\mathcal{E}_2$ . This is analogous to residual interference – interference that has been partially, but not completely, nulled out – due to other users in a communication system. However, this interference does not impact the localization process much since such “switches” in ToAs only occur when the ToAs themselves are close to each other. Specifically, the errors on these occasions are only  $\{11, 5.53, 5.37, 4.42\}$  meters larger than that observed with the genie-based scheme, which is acceptable for most purposes.

## VIII. CONCLUSIONS

The problem of matching observations to events is a fundamental bottleneck in space-time localization of multiple events using ToA sensors. The proposed framework sidesteps this bottleneck by a neuro-inspired first stage which generates a circular region as the “spatial response” of a particular ToA

at a particular sensor to a hypothesized event occurrence time. This enables us to quickly produce a reasonably good set of candidate events, which can subsequently be refined using statistical measures of goodness. We are therefore able to postpone addressing the matching problem to a point at which the list of candidate events is relatively small, with accurate space-time estimates for the true events in the list, at which point a linear program relaxation is effective in pruning out phantom events and outliers, and in handling misses. We believe that this general approach of treating sensor readings as generating responses in space-time can be generalized to almost any combination of sensors: for example, an angle-of-arrival sensor with a given angular error generates a cone, and a binary proximity sensor [10] generates a circle, over a window of event occurrence that depends on the sensing range and the time resolution. Thus, an important topic for future research is to extend and evaluate our framework for space-time localization using heterogeneous sensors. Another interesting area of investigation is to understand the fundamental limits of space-time localization, including statistical limits on accuracy, as well as characterizing when the matching problem is solvable.

## IX. ACKNOWLEDGMENT

Research 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.

## 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 Sensys 2004*.
- [2] D. Hubel and T. Wiesel, “Brain mechanisms of vision.” *Scientific American*, vol. 241, no. 3, p. 150, 1979.
- [3] X. Wang, Z. Wang, and B. O’Dea, “A toa-based location algorithm reducing the errors due to non-line-of-sight (nlos) propagation,” *Vehicular Technology, IEEE Transactions on*, vol. 52, no. 1, pp. 112–116, 2003.
- [4] M. Gavish and A. Weiss, “Performance analysis of bearing-only target location algorithms,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 28, no. 3, pp. 817–828, 1992.
- [5] Y. Chan and K. Ho, “A simple and efficient estimator for hyperbolic location,” *Signal Processing, IEEE Trans.*, vol. 42, no. 8, 1994.
- [6] A. Beck, P. Stoica, and J. Li, “Exact and approximate solutions of source localization problems,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 5, pp. 1770–1778, 2008.
- [7] A. Bishop, B. Fidan, K. Dogancay, B. Anderson, and P. Pathirana, “Exploiting geometry for improved hybrid aoa/tdoa-based localization,” *Signal Processing*, vol. 88, no. 7, pp. 1775–1791, 2008.
- [8] J. Chen, R. Hudson, and K. Yao, “Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field,” *Signal Processing, IEEE Transactions on*, vol. 50, no. 8, pp. 1843–1854, 2002.
- [9] G. Mao, B. Fidan, and B. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, vol. 51, no. 10, 2007.
- [10] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, “Target tracking with binary proximity sensors,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 4, p. 30, 2009.