Self-Localization Application for iPhone using only Ambient Sound Signals

Thomas Janson, Christian Schindelhauer, and Johannes Wendeberg
University of Freiburg, Department of Computer Science, Georges-Köhler-Allee 51, D-79110 Freiburg
{janson, schindel, wendeber}@informatik.uni-freiburg.de

***Abstract*—We present a smartphone application to localize a group of networking devices in a mobile environment without the need of any further infrastructure. Ambient sound signals are the only information source. Time marks are assigned to the recorded audio stream for distinctive audio events, out of which we evaluate the time differences of arrival (TDOA) between the devices. In contrast to common multilateration approaches we do not need any positional anchor points – neither any predefined smartphone positions nor the positions of the environmental sounds. As an application scenario we can localize arbitrary devices using only the random environmental noise peaks, e.g. in crowded areas like market places or concerts with the usual soundscape, or for thunderstorm tracking. Especially, our solution becomes useful when established positioning systems (e.g. GPS) are too imprecise or fail, as during indoor self-localization.**

We use a Wi-Fi connection to synchronize the clocks of the devices and to exchange time marks. In our experiments we evaluated the audio information and synchronized the devices up to an order of 0.1 ms. This results in a positioning precision in the order of 10 cm.

I. Introduction

Many applications on mobile devices like smartphones, PDAs, laptops, and tablet computers rely on position information, e.g. for filtering data according to the location context. Precise locality information is rarely provided by the communication network itself. A common approach is to equip the devices with additional hardware like GPS receivers which raises costs and increases the energy consumption. However, these approaches fail in shielded areas, in indoor environments and for small distances. Here, a common method is to create an infrastructure with fixed anchor points and to compute the positions in the communication network using time of arrival (TOA), time differences of arrival (TDOA) or the received signal strength indication (RSSI) of radio signals.

In our approach the devices read only ambient sounds like snapping fingers, clicking noises, coughing of nearby passengers or other sounds which are ubiquitously available. Based on the TDOA between the devices, the positions of the devices and the sound signals as well can be computed.

We use three methods that resolve the positions of the devices. Their application is dependent on the specific scenario, the number of devices available and the origin of the ambient sound signals. The Iterative Cone Alignment algorithm calculates the positions of both the devices and the sound signals in an iterative spring-mass simulation. It requires at least four devices connected in a network for the localization in the plane and five devices for three-dimensional positioning. The Ellipsoid TDOA method needs exactly three devices in the plane and four devices in three-dimensional space and assumes distant origins of the sound signals, also see Table I.

978-1-4244-5864-6/10/$26.00

If only two devices are available, merely a rough estimation can be made, based upon the given time differences of sound signals. We present the novel Arc Cosine Regression algorithm which estimates the distance between two devices in two-dimensional space. Compared to ordinary methods it achieves higher robustness and reliability.

Furthermore, we designed and implemented a prototypical localization framework currently available for laptop computers and iPhones. These mobile devices record ambient sound signals with their built-in microphones. They are connected via an arbitrary network, like Wi-Fi, GSM or the internet to synchronize their clocks and to exchange time marks. The computation is local on each device, and from the time marks received from the other microphones logical sound events are deduced and the relative positions are calculated and displayed. As the main criterion for the deduction of logical sound events the temporal proximity of time marks is used.

The localization methods use sound events as the only input to determine the positions of all devices and the location results are displayed on screen as an interactive 3D graphics visualization.

Our solution can replace existing positioning methods like GPS and wireless network localization when only relative localization is needed.

A. Related Work

The localization problem of mobile devices has been a broad and intensive research topic for infrastructural localization like GPS, Wi-Fi fingerprinting or GSM localization. A popular application of infrastructural localization is GSM localization of mobile phones. Various techniques exist, including angle/direction of arrival (AOA/DOA), time of arrival (TOA, "time of flight") and time difference of arrival (TDOA) [1]. Commonly a distance function using the received signal strength indication (RSSI) is used. Stable results can be achieved by fingerprinting using a map of base stations [2].

Another intensive research area is localization using the RSSI function of Wi-Fi signals. Methods include Bayesian inference [3], semidefinite programming for convex constraint functions [4], [5], a combination of Wi-Fi and ultrasound for TOA measurements like the Cricket system [6] or combina-

TABLE I

AVAILABLE METHODS FOR A MINIMUM NUMBER OF RECEIVERS.

Minimum number of receivers	In two dimensions	In three dimensions
Two	Distance estimation	
Three	Ellipsoid TDOA for a triangle	Distance estimation
Four	Iterative Cone Alignment	Ellipsoid TDOA for a tetrahedron
Five and more	Iterative Cone Alignment	

tions of methods [7]. TOA distance functions can be used [8], [9]. Wi-Fi Beamforming uses sensor arrays to determine signal directions [10], [11].

RSSI evaluation usually comes with difficulties for indoor localization due to the unpredictability of signal propagation [12]. We focus on TDOA analysis in our approach. For TDOA localization of sound and RF signals there is a basic scheme of four or more known sensors locating one signal source. This is solved in closed form [13], [14] or by iterative methods [15], [16], [17]. TDOA determination can be done by cross correlation of pairs of signals. An optimal shift between signals is calculated, corresponding to the angle of the signal [18]. However, we use signals with a characteristic peak.

Many approaches use pairs or sets of receivers to locate the direction of audio sources by TDOA evaluation. Valin et al. employ a grid of receivers and locate a speaker's direction [19]. Keyrouz and Diepold [20] simulate human binaural hearing. They locate the direction of a signal using TDOA and direction dependent signal characteristics of a human head dummy. Fletcher et al. locate a signal emitter on the hyperbola between two receivers [21].

B. Problem setting

Given a network of n synchronized devices $\mathbf{M}_i\,(i = 1,\ldots,n)$ at unknown positions in p-dimensional Euclidean space $\mathbb{R}^p$. Now m sound signals $\mathbf{S}_j\,(j = 1,\ldots,m)$ are produced at distant positions and at unknown times t_j. They travel with a fixed speed of sound c in a direct line. We expect that the signals are discrete in a manner that they can be distinguished from each other. Especially, we assume that we can distinguish the direct signal from echoes from surrounding walls. The signals are received by the devices at fixed time points $t_{i,j}$. This is the only input given in this problem setting.

Now the problem is to compute the positions of all receivers solely from the information provided by the signal times. Once the receiver positions have been calculated, all the signal positions can be calculated by hyperbolic multilateration.

Our findings concentrate on audio signals and the speed of sound. Of course, all of our findings can be applied to radio or light signals if the synchronization problem can be solved for the much higher speed of light.

The mathematical constraints are described by the signal propagation equation

$$c(t_j - t_{i,j}) = ||\mathbf{S}_j - \mathbf{M}_i|| \tag{1}$$

where $||\cdot||$ denotes the vector norm in Euclidean space.

By squaring the equations of form (1) we yield a quadratic equation system which can be written in quadratic form. Depending on the number of signals and receivers this system is under-defined, well-defined or even over-defined.

It can be rewritten as an optimization problem where a polynomial function of degree four needs to be minimized. There is only small hope for an efficient solution for such problems in general.

We have implemented an iterative method, the Cone Alignment, which solves the problem in many cases. It requires at least four receivers in two-dimensional space and five receivers in three-dimensional space [22].

However, if the signals origins can be assumed to be on the horizon, i.e. being in a far distance, the problem can be solved in an elegant closed form solution, requiring fewer receivers: The Ellipsoid TDOA method [23] deals with the problem of three unknown receivers in the plane. In this paper we briefly summarize this method.

Next, we present and discuss two methods which reconstruct the distance information between pairs of receivers using only time differences at the receivers: The robust variance method and the novel Arc Cosine Regression method which outperforms the variance method in robustness and precision and provides additional information about the uncertainty of the estimation. They are compared to a simple convex hull method based on the maximum measured time differences.

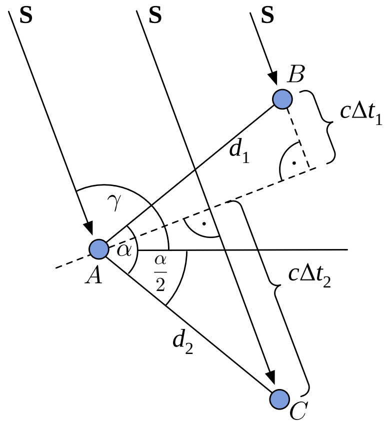
So, we have methods for two receivers, three receivers and for at least four receivers, which covers all feasible situations for relative positioning, see Table I: The Cone Alignment, the Ellipsoid Method and the distance estimation methods form the algorithmic background of an application for robust and mobile infrastructure-less localization.

II. METHODS

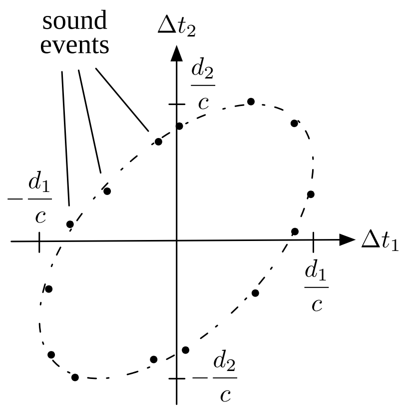
If less than four receivers are available in the plane then there is no unique solution to the problem of sound source localization, if the positions of the receivers are not known. However, under the assumption of infinitely distant sound sources three receivers suffice. The Ellipsoid TDOA method determines the relative distances in a triangle of devices [23]. We briefly summarize it in the following.

A. Ellipsoid TDOA method

Under the assumption that the sound signals originate from a distant location the time difference of signal reception between pairs of receivers depends only on the angle of the sound source. Given three receivers A, B and C a sound signal $\mathbf{S}$



(a) Three receivers A, B, C and a signal on the horizon with direction $\mathbf{S}$.



(b) Multiple distant signal sources with time difference pairs $(\Delta t_1, \Delta t_2)$ in two dimensions form an ellipse.

Fig. 1. Ellipsoid TDOA method

arrives at time points t_A, t_B and t_C, see Fig. 1(a). The time differences of arrival between the receivers are $\Delta t_1 = t_B - t_A$ and $\Delta t_2 = t_C - t_A$. For the angle $\alpha = \angle_{CAB}$ we choose the bisection of α and define γ as the angle of the sound source. Now we write:

$$x \quad := \quad \Delta t_1 = d_1 \cos(\gamma - \alpha/2) \tag{2}$$

$$y \quad := \quad \Delta t_2 = d_2 \cos(\gamma + \alpha/2) \tag{3}$$

Combining these equation leads to the following ellipse equation:

$$x^2 \frac{1}{d_1^2} + y^2 \frac{1}{d_2^2} + xy \frac{-2\cos\alpha}{d_1 d_2} = \underbrace{\frac{1}{2} - \frac{1}{2}\cos 2\alpha}_{\sin^2\alpha} \tag{4}$$

It is helpful to draw a diagram of sound events $\mathbf{S}_j$ using cartesian coordinates where the axes are Δt_1 and Δt_2. So, the points corresponding to all possible sound events form an ellipse with the origin as the center, see Fig. 1(b).

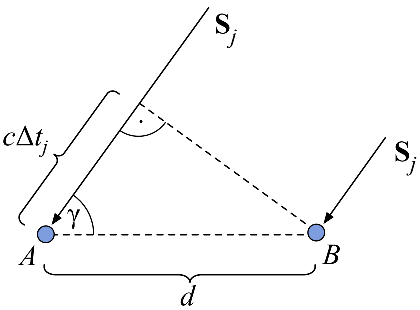


Fig. 2. Arc Cosine Regression method. A distant sound signal $\mathbf{S}_j$ arrives at A and B. Angle γ is calculated as $\cos(\gamma) = \frac{c\Delta t}{d}$.

From any set of three distinct sound events the ellipse equation (4) is uniquely determined, except for mirror symmetric solutions. Then, from the equation the values d_1, d_2 and α of the receiver triangle can be derived using basic algebra.

B. Arc Cosine Regression method

If we have only two receivers the Ellipsoid method can no longer be used. Still, we are trying to estimate the distance between the receivers.

Assuming a uniform distribution of the sound signals on the horizon, it is possible to estimate this distance. Given two receivers A and B in two-dimensional space (Fig. 2). A number of signal sources $\mathbf{S}_j$ are created at distant points in space arriving at times $t_{A,j}$ and $t_{B,j}$ at A and B, respectively. They originate from equally distributed angles γ described as

$$\cos(\gamma) = \frac{c\Delta t}{d} \tag{5}$$

where c is the speed of sound, d the unknown distance between A and B, and $\Delta t_j = t_{A,j} - t_{B,j}$ is the time difference of the signals arriving at both receivers.

We consider the upper semi-circle above A described by the interval $[0, \pi]$ and sound sources arriving in the circle sector $[\phi, \pi]$ for arbitrary $\phi \in [0, \pi]$. The probability that ϕ is greater than the angle γ of any given sound source is noted as

$$P(\phi > \gamma) = \frac{\phi}{\pi} \quad . \tag{6}$$

The lower semi-circle in $[0, -\pi]$ is symmetric. It combines to the same probability

$$P(|\phi| > |\gamma|) = \left|\frac{\phi}{2\pi}\right| + \left|\frac{-\phi}{2\pi}\right| = \frac{\phi}{\pi} \quad .$$

This is rewritten to the accumulated probability that Δt is greater than any given Δt_j:

$$F(\Delta t) = 1 - \frac{\phi}{\pi} = 1 - \frac{1}{\pi}\arccos\left(\frac{c\Delta t}{d}\right) \tag{7}$$

The accumulated probability function $F(\Delta t)$ is depicted in Fig. 3.

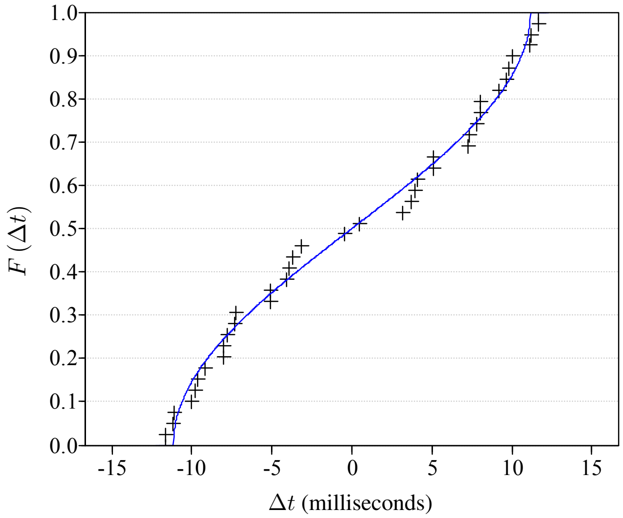


Fig. 3. Accumulated probability distribution $F(\Delta t)$. 20 time differences are duplicated and sorted by their value (black crosses). The blue curve is the regression of an arc cosine curve of the values. The maximum time difference is $\Delta t = \pm 11.7\,\mathrm{ms}$.

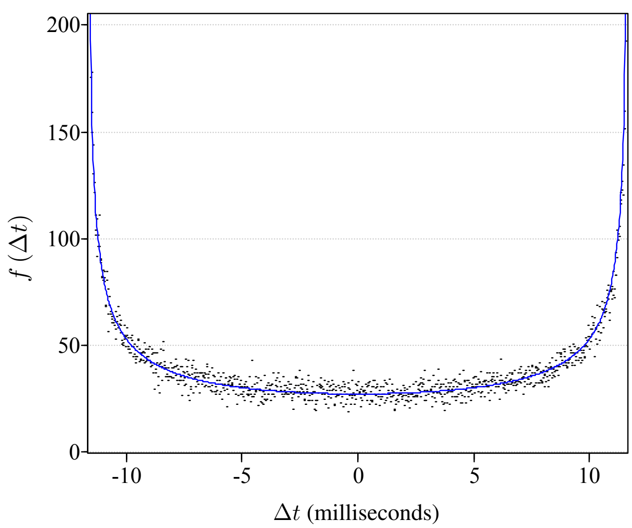


Fig. 4. Density function $f(\Delta t)$ (blue curve). The black dots indicate the normalized count per interval for a random sampling of 100 000 equally distributed distant sound sources.

The derivative of F yields the density function

$$f(\Delta t) = \frac{1}{\pi\sqrt{\frac{d^2}{c^2} - \Delta t^2}} \tag{8}$$

which is defined in the domain $\left(-\frac{d}{c}, \frac{d}{c}\right)$. A sampling of 100 000 distant sound sources was performed and assigned to intervals of time differences. The normalized count per interval reflects the density function from an experimental point of view. See Fig. 4 for an illustration of the density function and the sampling experiment.

A common method to estimate the distance between two receivers is the evaluation of the variance of the time differences of arrival. Given a continuous probability density function $f(x)$ the variance of a random variable X is described as

$$\mathrm{Var}(X) = \int (x-\mu)^2 f(x)\,\mathrm{d}x \enspace .$$

Under the assumption that the signals are equally distributed on the horizon and the receivers are synchronized we state the mean μ to be zero. Then, the variance of the time differences is evaluated to

$$\int_{-\frac{d}{c}}^{\frac{d}{c}} x^2 f(x)\,\mathrm{d}x \;=\; \sigma^2 = \frac{d^2}{2c^2} \enspace . \tag{9}$$

Eq. (9) is solved for the receiver distance $d = c\sqrt{2\sigma^2}$. We call this the *variance method*.

As noted, the clocks of the receivers are synchronized. Then, the time differences are symmetric: For every Δt there exists $-\Delta t$, a sound signal originating from the opposite direction. We duplicate the time differences Δt by their negative pendants improving the variance estimation by this symmetry.

A more sophisticated technique uses the accumulated probability distribution to gather information about the receiver distance d, the *Arc Cosine Regression method*. The goal is to solve the anti-derivative $F(\Delta t)$ for d. Again we use symmetric duplication of the time differences by their negative pendants. The time differences are sorted by their value Δt_j and assigned a coordinate $(\Delta t_j, \frac{1}{m} I_j)$ where I_j denotes the sorted index $0 \le I_j \le m$. For a minimum number of $m \ge 1$ sound signals we use linear regression. It can be applied after we rewrite Eq. (7) such that d appears as a linear coefficient. We know that $\sin(\arccos(x)) = \sqrt{1-x^2}$, so we write:

$$\begin{aligned} I_j &= 1 - \tfrac{1}{\pi}\arccos\left(\tfrac{c}{d}\Delta t_j\right) \\ \Rightarrow \underbrace{\left(1 - \sin\left((1-I_j)\pi\right)^2\right)}_{\mathbf{A}} \underbrace{d^2}_{\mathbf{x}} &= \underbrace{\Delta t_j^2 c^2}_{\mathbf{b}} \end{aligned}$$

Now we solve the equation system $\mathbf{Ax} = \mathbf{b}$ using the least squares method:

$$\begin{aligned} \left(\mathbf{A}^T\mathbf{A}\right)\mathbf{x} &= \mathbf{A}^T\mathbf{b} \\ \mathbf{x} &= \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\left(\mathbf{A}^T\mathbf{b}\right) \end{aligned} \tag{10}$$

With the vector $\mathbf{x} = \{x_1\}$ we finally retrieve $d = \sqrt{x_1}$.

One may argue that in realistic environments a situation of equally distributed sound sources will hardly be seen. Our simulations described in the following section point out that the regression renders the results robust and the assumption not at all far-fetched, especially for increased numbers of sound events. Anyhow, the method also appears suitable for small numbers of sound events.

C. Simulation

We have tested the distance approximation algorithms in a simulated environment using a numerical computing software. For the choice of parameters we concentrate on sound signals.

The two algorithms, the Arc Cosine and the variance method, are compared to a simple convex hull method, which deduces the distance d from the limits of the occurring time differences:

$$d = \frac{c}{2}\left(\Delta t_{\max} - \Delta t_{\min}\right) \quad (11)$$

Again, the method uses duplicated time differences as described for the methods before.

Two microphones are placed in two-dimensional Euclidean space with a distance of 4 m between them. This is a typical real-world value and can be assumed without loss of generality, as scaling the microphone distance is contained in scaling the distance of the sound sources and signal runtime errors.

Sound sources are placed at random positions on a circle around the center of the two microphones. The times when sound signals arrive at the microphones are calculated, given a signal speed of c = 343 m/s.

The behavior of the approximation is examined under four types of parameter variation: We vary the distance of the sound sources, the number of sound sources and the influence of runtime errors during signal reception. The error model is created as a Gaussian distribution which is added when signals are timestamped at the receiver. As the last parameter we violate the assumption of equally distributed sound sources on a circle around the microphones.

a) Sound source distance: We assume that sound signals originate from far distances. However, the distance approximation method can also be applied for medium-ranged scenarios with some deterioration of performance.

We simulate 20 sound sources residing at random positions on concentric circles with radii of 0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 10, 12, 15, and 18 meters around the microphones. For every radius we generate series of 100 runs and execute the algorithms. No runtime error is assumed in this test. Fig. 5 shows the the mean and standard deviation of the 100 runs for every radius.

We observe a systematic underestimation for close-ranged distances. This is no surprise – when sound sources are enclosed by the microphones the maximum time differences of arrival cannot occur and the regression yields a narrow arc cosine curve. For the variance and hull method we observe the same behavior. The results recover if the sound sources are at least at an equal range as the microphones, which is a radius of 2 m.

These results indicate that we only need sound sources slightly further away from the center than the microphones – instead of the assumed distant sound sources.

b) Number of sound sources: In a second test we set the sound sources to a fixed radius of 20 m and vary the number of sound signals. We start with one source and run the test one hundred times. This leads to a underestimation of the distance when the time difference is below the possible maximum.

When we duplicate the time differences, as described in the previous section, the variance can be calculated even for one single sound event. The variance method averages to the

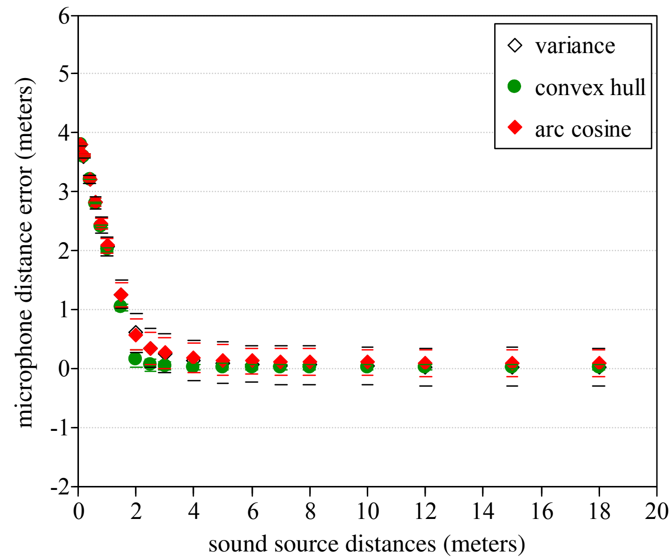


Fig. 5. Increasing sound source distance results in low approximation errors. Good results can be seen at radii of 2 m and above.

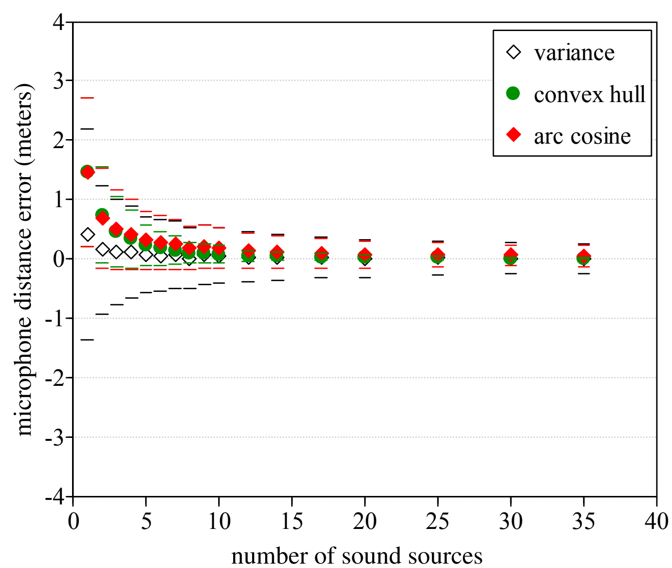


Fig. 6. Higher numbers of sound sources result in better approximation results. The variance method performs better when the number of sound signals is small, though with a high standard deviation.

smallest underestimation of all methods – with a huge standard deviation, though.

With two sources the distance error of the Arc Cosine method decreases to 0.68 m, which is less than 20 % of the microphone distance of 4 m. If more than five sound signals are available the results become stable, which is indicated by the decreasing standard deviation of the test series. See Fig. 6 for the graphical illustration of mean and standard deviation.

c) Runtime errors: The third experiment examines the behavior of the algorithms under tightened conditions. The time differences of arrival at the receiver can no longer be precisely measured but vary by a Gaussian distributed error.

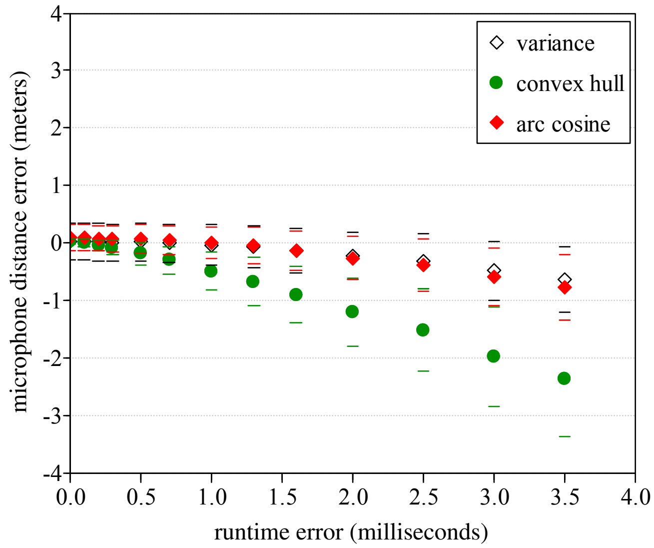


Fig. 7. Gaussian runtime errors lead to an over-estimation of the distance and high variance between runs. Realistic runtime errors lie below 1.0 ms.

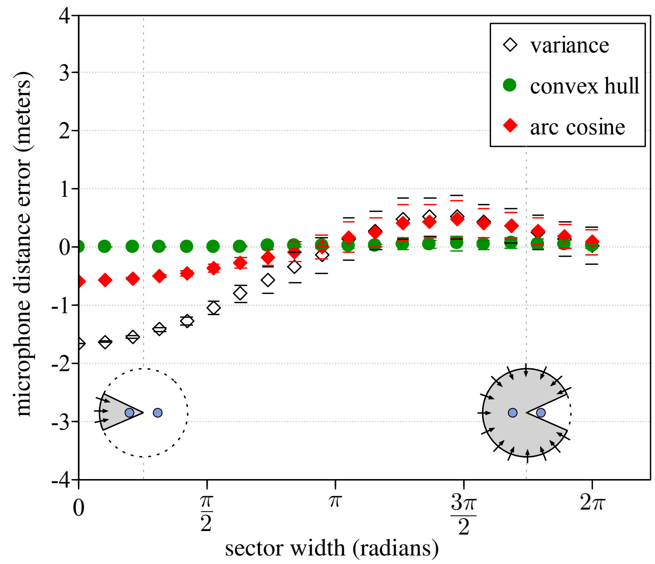


Fig. 8. Sounds from an increasingly smaller sector violate the assumption of equal distribution of sound sources. Then, especially the variance method suffers from false estimation.

This adds realism to the simulation: Timing errors originate from bad synchronization of the receivers, from imprecise audio processing or from varying hardware latency. Runtime variations could also be effected by wind or by air temperature differences.

Our experiment uses 20 sound sources. We start with a slight error, a standard deviation of the runtime of 0.1 ms and increase to a high error of 3.5 ms.

A moderate overestimation and an increase in standard deviation appears in all three methods, while the results of the variance and the Arc Cosine methods stay plausible, altogether, see Fig. 7. In contrast, the hull method has a severe overestimation: Together with the runtime errors the maximum possible time differences increase.

A test run with a reduced number of 8 sound sources leads to similar results with a slightly increased standard deviation.

In our real-world experiments with the Apple iPhones we encounter a standard deviation of the runtime error below 1 ms.

d) Sector width: Of great interest is the behavior of the algorithms if the assumption of equally distributed sound sources is violated. We simulate a cutout of sound sources: The signals originate from an increasingly narrow sector, such that sound signals come only from a single direction. Again, we use 20 sound sources at a distance of 20 meters.

In the case of a "vertical" cutout the last remaining sound sources reside on the line of no time difference between the microphones. The occurring time differences are near to zero. In this case all three algorithms fail as they cannot distinguish the small TDOA resulting from a narrow angle or from a small receiver distance.

In the "horizontal" case the last remaining sound sources reside on a line with the microphones, while the occurring time differences are close to the extrema. Then, the variance method tends to immense overestimation, while the Arc Cosine method still reports reasonable values, see Fig. 8.

The hull method is insensitive to this scenario, because the maximum values do not change. However, it is disqualified if runtime errors occur, which is common in reality.

The result of this last simulation suggests the employment of the Arc Cosine method due to its robustness in the case of runtime errors and when the sounds are not perfectly distributed.

III. Smartphone Application

We have designed and implemented a prototypical software framework to support the development of TDOA localization applications. Modules for network connection and management, time synchronization and audio processing are included, as well as an interface for 3D visualization. The framework can serve as a platform for the verification of our algorithms and for the programming of neat and easy-to-use gadget applications for the public domain. Fig. 9 overviews the components of the framework.

We assume that several devices are connected in a network and are able to communicate via TCP or UDP. Then, the devices find out their vicinity in the local network and the *Network Manager* component maintains a routing table of all involved devices. Whenever a sound signal is detected by a device its timestamp is forwarded to the other devices where it is assembled to a sound event as input for the localization.

Localization based upon time differences relies on precise synchronization among the devices. Unsynchronized localization is generally possible, but the time offset and the drift between clocks are additional system variables which increase the number of required sound events to determine them.

We use synchronized localization in our algorithms which is provided by the *Time Synchronization* component. One unique master is chosen among the devices which serves

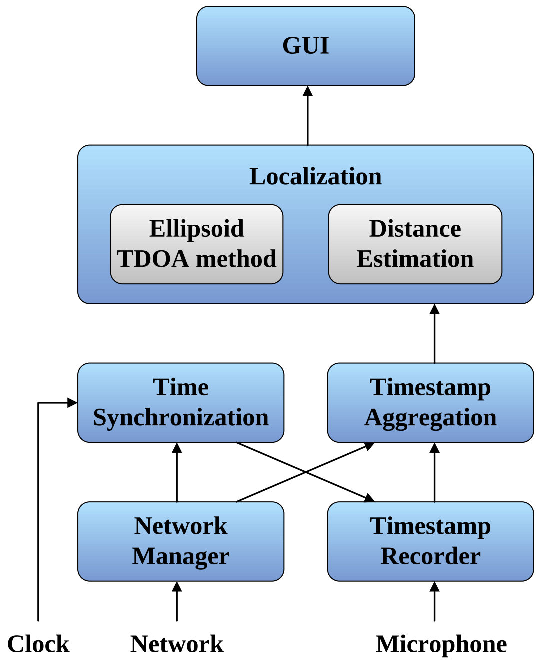


Fig. 9. Components of the software framework. Synchronized timestamps are recorded on every device and routed to the other devices. There, they are aggregated and forwarded as input for the TDOA localization algorithms.

as the time reference. All other devices periodically query the master for its current time. This is done by a series of pinging the master which answers with its current time T_{ref}. This timestamp is corrected by 1/2 RTT (round trip time), assuming the network packet took the same runtime in both directions. The obtained timestamps are filtered for high RTT, which result from network jitter.

Our experiments indicate that clock drift correction is essential even with the utilized high precision event timers (HPET). Although running with accurately constant speed, drift rates between HPET clocks of 0.03 % were observed, which is very high. We obtain both reference time T_{ref} and clock drift between client and master by linear regression of the timestamp set. The client clock is corrected by the clock drift rate τ_{drift}. The synchronized time on all devices is

$$T_{\text{sync}} = T_{\text{ref}} + \frac{1}{2}t_{\text{RTT}} + (T_{\text{client}} - T_{\text{last}}) \cdot \tau_{\text{drift}} \qquad (12)$$

where T_{last} denotes the time of the last synchronization and T_{client} denotes the local time of the client.

The *Timestamp Recorder* component reads the microphone and searches the audio stream for distinctive audio events. The audio track is filtered for sharp sound events, like clapping or finger clicking, and their points in time are determined. As a peculiar mark for a sound event we use the moment when the signal rises above a environment noise dependent threshold for the first time. Here, background noise is filtered implicitly. Fig. 10 displays an example for hands clapping.

Threshold comparisons turned out to be the most robust approach with only little drawbacks in precision. Maximum searches, either directly or derivative (edge detection) showed

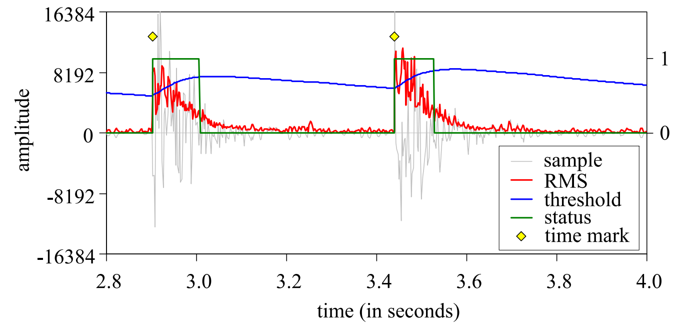


Fig. 10. Timestamp Recorder. Environment noise dependent threshold analysis of hands clapping. The moment when the signal rises above the threshold is chosen as the timestamp.

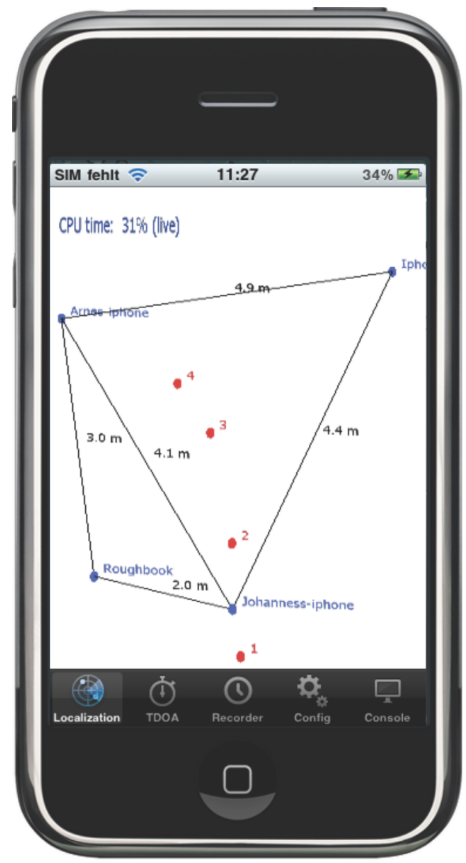


Fig. 11. Screenshot of our application running on an Apple iPhone 3GS. The screen displays the positions of four receivers (blue dots) and four sound signals (numbered red dots) in the plane as a view from above.

to be slightly more precise but proved to be ambiguous with fatal results in cases when hosts chose different maxima. In the case of clicking audio signals with a steep initial edge we calculate time marks with a precision of 0.1 ms. The spatial equivalent is 3 cm. However, when signals are smooth, like human speech, the time point of signal detection is not determined. Clicking fingers or clapping one's hands is still adequate. In the following experiments we use sharp signals created by clapping two wooden planks.

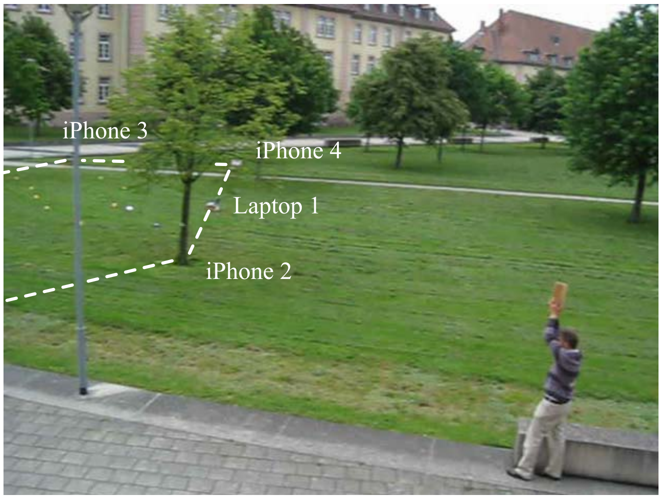


Fig. 12. Four iPhones and four laptops to be located from a series of distant sound signals. The experimenter in the picture claps the wooden planks, compare the device positions in Fig. 14.

The *Timestamp Aggregation* component collects the local time marks from the Timestamp Recorder and the remote time marks from other devices in the network. They are combined to logical sound events. It is essential to aggregate the time marks correctly to one sound event from the real world. We choose temporal proximity of timestamps as the criterion. We assume that the time between sound events is greater than the maximum TDOA in a sound event, such that the aggregation is uniquely determinable. Otherwise, we filter ambiguous events to prevent false assembling.

Based on the framework we have implemented an application for laptop computers and the Apple iPhone. The iPhone 3GS provides all necessary hardware for our project: A precise hardware timer (HPET), a Wi-Fi connection, a high quality audio interface, and hardware accelerated 3D graphics. Fig. 11 shows a screenshot of four computers located in a two-dimensional environment. The positions of the devices have been calculated, as well as the positions of four sound events.

IV. Real-world experiments

We have tested the theoretical approaches in several real-world experiments. We use a network of laptops and Apple iPhones as network nodes. Our software establishes network communication and provides precise time synchronization. With the built-in microphones we record sound signals. The signals are exchanged to every participating computer and computed locally.

The tests took place on a green field on our campus. Four laptops and four iPhones were connected in a wireless network using a Wi-Fi access point. Of course, one of the laptops could have acted as a Wi-Fi hotspot as well. The network nodes were positioned in an ellipse-like formation of the size $30\,\text{m} \times 25\,\text{m}$.

A volunteer agreed to act as a noisemaker utilizing two wooden planks which were clapped. He was free to choose positions making noise, but he was advised to choose varying

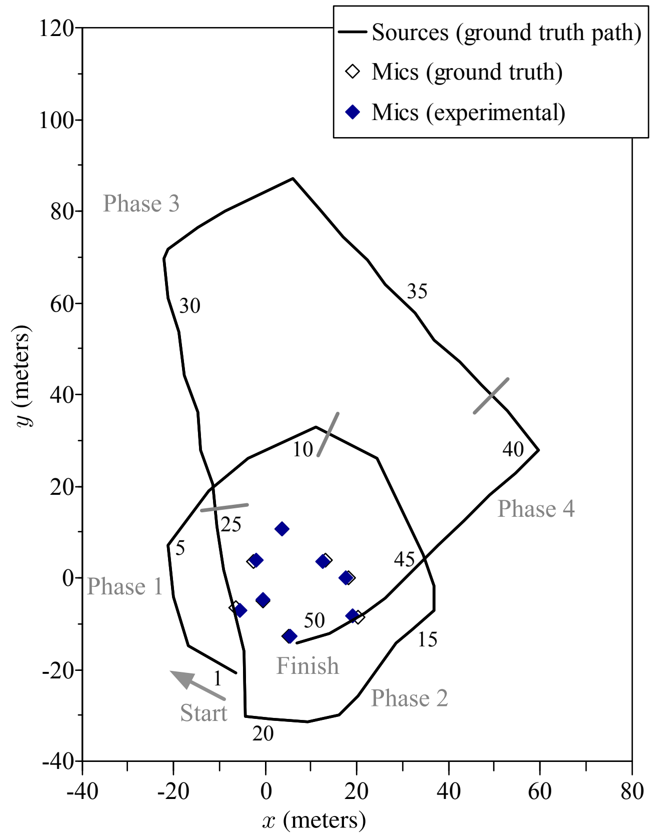


Fig. 13. Overview on the microphone positions. The black line is the experimenter's path around the field. The four phases describe the experiment development, see Fig. 15.

locations. A series of more than 50 sound signals was created with sound signals at every 10–20 meters.

We noted down the positions of all laptops and smartphones by measuring the distance to two anchor points. The anchor points were chosen as a reference for a rectangular coordinate system. Then, the coordinates of the devices were calculated using trilateration. The positions of the microphones were charted up to a precision of 10 cm. The trajectory of the volunteer was recorded on video and transcribed to a polygon path using world coordinates. We used an aerial image of the green field and video recordings of the experiment to retrieve the path of the experimenter (Fig. 13). The positions of the sound signals were determined according to their geographic coordinates up to a precision of 5 m.

The sound events recorded during the tests were assigned a timestamp. The Arc Cosine Regression algorithm got these timestamps as the only input and computed the distances between all computers and iPhones. They form a complete graph of $\frac{1}{2}(n-1)n = 28$ distances between nodes. Using optimization we calculated the relative positions (x_i, y_i) of the microphones from the distances d_{ij}:

$$\min_{x,y} \left(\sum_{i=1}^{n} \sum_{j=i+1}^{n} (x_i - x_j)^2 + (y_i - y_j)^2 - d_{ij}^2 \right)$$

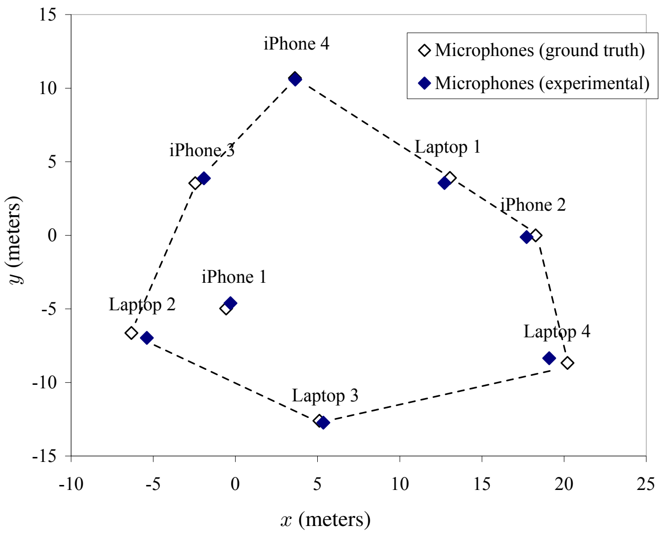


Fig. 14. Large-scale view of the microphone positions. The average position error of the microphones is 0.58 m with a standard deviation of 0.35 m.

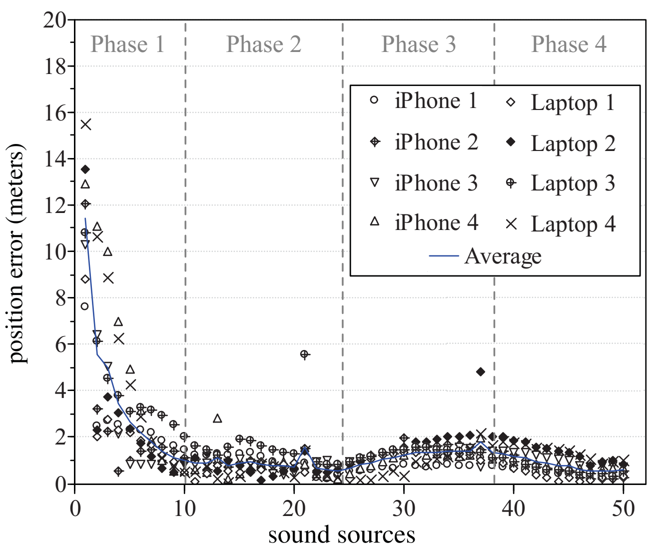


Fig. 15. Sound sources are added consecutively, beginning with a single source. With more than six sources the position error has decreased rapidly.

The resulting positions were mapped to the real-world positions by a congruent rotation and translation. This was done by calculating the SVD (Singular Value Decomposition) of the point set correlation which provides a transformation to minimize deviation between estimated and real positions in the least squares sense.

The comparison of the localization algorithm and the measured microphone positions revealed an average deviation of 0.58 m with a standard deviation of 0.35 m. Fig. 14 shows a large-scale view of the real microphone positions and the results of our estimation.

In a subsequent evaluation the sound sources were added consecutively starting with a single sound signal. As the experimenter progressed on his track making noise, more sounds were added and the calculation was repeated with the new information. The obtained microphone distances were merged into relative coordinates as in the experiment before and fit to the real positions. The progress of the position errors for every receiver and for every number of sound sources is illustrated in Fig. 15.

Position errors decrease quickly after more than six sound sources have been added. This has to be attributed not only to the number of sound sources but also to the path covered by the experimenter (compare Phase 1 in Fig. 13 and 15): Only when the path describes a semi-circle around the microphones their positions can be discovered. Then, errors reside below a maximum of 2 m with an average of about 1 m – throughout an acceptable value in a setup of an edge length of 30 m.

Note the error increase in Phase 3. This is when the experimenter's distance increases and the signals arrive predominantly from the same direction. This violates our assumption of equally distributed sound sources resulting in a moderate error increase to a maximum of 3 m.

Some outliers must be noted, for example the position of Laptop 2 with 37 sound sources. This is a result of deficient position reconstruction of the distances and not directly a problem of the distance estimation. The optimization algorithm might have run into a local minimum and estimated a microphone entirely wrong. This can be solved by repeating with different initial values.

V. Conclusions

We have considered the problem of relative localization of nodes in a computer network solely based on ambient signals. There is absolutely no knowledge available about the received audio signals except that they can be distinguished from each other. There are no anchor points given in the network. Of course, a provided anchor point can extend the relative localization to an absolute one. We use the fact that the network nodes are synchronized and that the ambient signals originate from punctual origins and we assume that they travel with a constant speed on a direct line. Then, we can evaluate the time differences of arrival to reconstruct the positions of the receivers.

For this problem we presented an elegant closed-form solution – the Ellipsoid TDOA method [23] for three receivers. Although only three receivers are given, thus rendering the underlying equation system under-determined, we provide a solution in the plane to reconstruct the receiver positions. We only need the assumption that the sounds originate from far away.

In this paper, we presented a method to estimate distances between only two receivers under the additional assumption that sound sources are equally distributed. It is based on the distribution of the time differences such that the distance can be recovered by a regression of the distribution function.

Simulations indicate that our Arc Cosine Regression algorithm outperforms a variance evaluation, especially when the assumption of equally distributed sound sources is violated. At the same time it retains the robustness of a variance method when measurement errors occur. This is in contrast to calculating the convex hull of the time differences and deducing the receiver distance out of it, which is prone to outliers.

We have created a software platform and implemented our algorithms. We installed the software on the Apple iPhone which provides the computational power and usability for highly mobile applications at the same time.

The practicality of our algorithms has been proven in a real-world experiment. A volunteer produced sound signals from positions of his choice using two wooden planks. With the built-in microphones the notebooks and smartphones nearby were able to compute their relative positions within an error below one meter.

A. Future work

It is very obvious that the distance estimation method using the Arc Cosine Regression can be employed to support the Iterative Cone Alignment as an initial guess of the iteration. The distance estimation might accelerate the approximation convergence and help to recover from ambiguities.

The Arc Cosine scheme could support the localization in a second way. As the regression is very robust against measurement errors it could be used to prefilter timestamps for the iterative approach, e.g. false aggregations of timestamps could be detected.

Furthermore, we plan to include the use of non-discrete continuous signals, e.g. voices, traffic noise or analogous radio signals. By testing for correlation of audio signals it should be possible to detect time differences analogously to sharp signals. This would dramatically increase the information basis of the algorithms.

Of importance is also the question of unsynchronized localization. The use of radio signals from Wi-Fi access points, from GPS or from broadcast prevents precise synchronization among receivers due to the greater speed of light. We will extend the distance estimation to unsynchronized operation.

Another interesting research topic is the mobility of continuous signal sources and of receivers. Filtering techniques like the Kalman Filter or particle filtering could be implemented to combine single measurements and track devices and sound sources over time.

References

[1] Christopher Drane, Malcolm Macnaughtan, and Craig Scott. Positioning GSM Telephones. *IEEE Communications Magazine*, 36:46–54, 1998.

[2] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal de Lara. Accurate GSM Indoor Localization. In *Ubicomp*, pages 141–158, 2005.

[3] Mihail L. Sichitiu and Vaidyanathan Ramadurai. Localization of Wireless Sensor Networks with a Mobile Beacon. In *Proceedings of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems*, pages 174–183, 2004.

[4] Pratik Biswas and Yinyu Ye. Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54, New York, NY, USA, 2004. ACM.

[5] Lance Doherty, Kristofer S. J. Pister, and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663, 2001.

[6] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, 2000.

[7] Zhonghai Wang and Seyed A. Zekavat. A Novel Semidistributed Localization Via Multinode TOA-DOA Fusion. *IEEE Transactions on Vehicular Technology*, 58(7):3426–3435, 2009.

[8] Nissanka B. Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller. Poster abstract: anchor-free distributed localization in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 340–341, New York, NY, USA, 2003. ACM.

[9] Srdjan Capkun, Maher Hamdi, and Jean Pierre Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, page 10 pp., Jan. 2001.

[10] Chunwei Jethro Lam and Andrew C. Singer. Bayesian Beamforming for DOA Uncertainty: Theory and Implementation. *IEEE Transactions on Signal Processing*, 54(11):4435–4445, Nov. 2006.

[11] Joe C. Chen, Kung Yao, and Ralph E. Hudson. Acoustic Source Localization and Beamforming: Theory and Practice. *EURASIP Journal on Applied Signal Processing*, 2003(4):359–370, 2003.

[12] Brian Ferris, Dirk Hähnel, and Dieter Fox. Gaussian Processes for Signal Strength-Based Location Estimation. In *Proceedings of Robotics: Science and Systems Conference (RSS)*, 2006.

[13] Le Yang and K. C. Ho. An Approximately Efficient TDOA Localization Algorithm in Closed-Form for Locating Multiple Disjoint Sources With Erroneous Sensor Positions. *IEEE Transactions on Signal Processing*, 57:4598–4615, Dec. 2009.

[14] M.D. Gillette and H.F. Silverman. A Linear Closed-Form Algorithm for Source Localization From Time-Differences of Arrival. *Signal Processing Letters, IEEE*, 15:1–4, 2008.

[15] Wade H. Foy. Position-Location Solutions by Taylor-Series Estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-12(2):187–194, March 1976.

[16] Don J. Torrieri. Statistical Theory of Passive Location Systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-20(2):183–198, March 1984.

[17] Dragana Carevic. Automatic Estimation of Multiple Target Positions and Velocities Using Passive TDOA Measurements of Transients. *IEEE Transactions on Signal Processing*, 55:424–436, Feb. 2007.

[18] Yong Rui and Dinei Florencio. New direct approaches to robust sound source localization. In *Proc. of IEEE ICME 2003*, pages 6–9. IEEE, 2003.

[19] Jean-Marc Valin, François Michaud, Jean Rouat, and Dominic Létourneau. Robust Sound Source Localization Using a Microphone Array on a Mobile Robot. In *Proceedings International Conference on Intelligent Robots and Systems (IROS)*, pages 1228–1233, 2003.

[20] Fakheredine Keyrouz and Klaus Diepold. An Enhanced Binaural 3D Sound Localization Algorithm. In *2006 IEEE International Symposium on Signal Processing and Information Technology*, pages 662–665, Aug. 2006.

[21] F. Fletcher, B. Ristic, and D. Musicki. Recursive estimation of emitter location using TDOA measurements from two UAVs. In *10th International Conference on Information Fusion*, pages 1 – 8, 2007.

[22] Johannes Wendeberg. Laufzeitbasierte Schallortung mit unbekannten Sender- und Empfängerpositionen. Master's thesis, University of Freiburg, 2009.

[23] Thomas Janson, Christian Schindelhauer, and Johannes Wendeberg. Self-localization Based on Ambient Signals. In *Algorithms for Sensor Systems*, volume 6451 of *Lecture Notes in Computer Science*, pages 176–188. Springer, 2010.