Journal of Location Based Services
Vol. 00, No. 00, Month 200x, 1–20

Calibration-Free TDOA Self-Localization

Johannes Wendeberg, Christian Schindelhauer
Department of Computer Science
University of Freiburg, Germany
{wendeber, schindel}@informatik.uni-freiburg.de

Fabian Höflinger, Leonhard Reindl
Department of Microsystems Engineering
University of Freiburg, Germany
{fabian.hoeflinger, reindl}@imtek.uni-freiburg.de

(*Draft – published in Journal of Location Based Services, May 2013*)

We present an approach for the localization of passive receiver nodes in a communication network. The only source of information is the time when environmental sound or ultrasound signals are received. The discrete signals occur at unknown positions and times, but they can be distinguished. The clocks of the receivers are synchronized, so the time differences of arrival (TDOA) of the signals can be computed. The goal is to determine the relative positions of all receiver nodes and implicitly the positions and times of the environmental signals.

Our proposed approach, the Cone Alignment algorithm, solves iteratively a non-linear optimization problem of time differences of arrival by using a physical spring-mass simulation. We present a geometrical representation of the error function, which is modeled by physical springs. By iterative relaxation of the springs the error function is minimized.

The approach is tested in numerous simulations, where our algorithm shows a smaller tendency to get stuck in local minima than a non-linear least-squares approach using gradient descent. In experiments in a real-world setting we demonstrate and evaluate a tracking system for a moving ultrasound beacon without the need to initially calibrate the positions of the receivers. Using our algorithm, we estimate the trajectory of a moving model train and of a RC car with a precision in the range of few centimeters.

Keywords: TDOA, self-calibration, localization, ultrasound

1 Introduction

The increasing availability and computational power of smartphones and handheld computers permits applications never before possible. Yet the exact position of the phone remains subject to external infrastructures like the GPS system, GSM multilateration, or Wi-Fi based location services.

Localization using these infrastructures depends on the availability of the external systems. Infrastructures might fail due to environmental conditions (indoor locations, in the forest, on mountains), temporal unavailability (network breakdown), or they could be deactivated for political reasons. Besides, these location services are mostly too imprecise, especially for indoor localization. Applications like navigation or augmented reality could benefit from precise location information.

We address the problem of self-localization of four or more receivers using the time differences of arrival (TDOA) of acoustic signals from the environment – of which we do not know the positions of origin. The benefit of TDOA multilateration is the absence of control over the sender. Any kind of signal can be used, if it can be distinguished, and if the time can be calculated when the signal is received by a group of receivers.

A sound source could be “natural”, like a finger snapping, coughing, or the tick sound of a metronome. In this way, our approach is completely passive, listening to signals from the environment. It may also use active signaling, requiring just a simple and inexpensive synthetic signal beacon. All we assume is, that the sound travels in a straight line from the signal source to the receiver and that we can distinguish the signal from the background noise.

In this paper, we present our novel *Cone Alignment* algorithm to calculate a solution to the TDOA self-localization problem using an interative spring-mass simulation. We propose a geometrical representation of localization error and we present an iterative approach to minimize the error.

Extended release of: J. Wendeberg, F. Höflinger, C. Schindelhauer, and L. Reindl. Anchor-free TDOA Self-Localization. In *Proc. of 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011.

Journal of Location Based Services
ISSN 1748-9725 print / ISSN 1748-9733 online © 200x Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/1748972YYxxxxxxxx

In numerical simulations we analyze the problem of local minima of iterative optimization and we show that our proposed approach can find a correct solution to a randomized problem setting with higher probability than an ordinary iterative minimization approach using gradient descent.

In real-world experiments we demonstrate the feasibility of our approach by locating a set of receivers using natural sounds from unknown positions. Furthermore, we present a calibration-free tracking system for a moving ultrasound sender, where the positions of ultrasound receivers do not need to be calibrated by hand. We show that we can locate the position of a sender using our proposed algorithm up to a precision of 5 cm.

1.1 *Related work*

Positioning of mobile devices with given infrastructure is a broad and intensive research topic. Popular infrastructure-based approaches for indoor and outdoor applications are GSM localization [1, 2] and Wi-Fi network fingerprinting [3]. The interpretation of the received signal strength indication (RSSI) is an usual approach [4].

When RSSI or TOA (time of arrival, "time of flight") data is available the problem is reduced to a problem of distance vectors. It is solved using the iterative Gauss-Newton method [5] or by linear estimators [6]. Force-directed approaches are an alternative relaxing distance constraints in large-scale networks [7, 8] and in the Vivaldi network coordinate system [9].

TOA/TDOA measurements using audible sound can be obtained by detection of discrete signals [10, 11] or by cross correlation of signals [12, 13]. Ultrasound is used in [14, 15].

We focus on TDOA localization. In most TDOA approaches the receivers' positions are given a priori. Then, estimating a sender's position using time differences of arrival can be addressed in closed form equations [16, 17] or by iterative approaches [18, 19]. Hongyang et al. use three anchor beacons [20]. Moses et al. use TDOA with additional angle information to locate unknown sender and receiver positions [21]. This would require expensive receiver arrays or directed receivers.

Localization without anchors and relying only on TDOA can be solved if assumptions on the signal positions are made, i.e. the signals originate from far away [11, 22, 23], which can be solved even if the receivers are not synchronized [24, 25]. Or they originate from at least outside, on the axis, of a pair of receivers [13, 26, 27].

Close to our problem setting is the approach of Biswas and Thrun [10]. No assumptions of the signal positions are required and only TDOA information is used to iteratively refine a Bayesian network using gradient descent. However, the correct solution cannot be found in every case. An upper error bound with signals in the unit disc is shown in [27].

A very elegant approach was proposed by Pollefeys and Nister [28]. The special case of ten or more receivers is solved in a linear approach without initialization and without assumptions on the positions. However, the approach seems to be prone to TDOA errors, and the minimum cases are not covered.

1.2 *Problem setting*

Consider a network of n receivers at unknown positions $\mathbf{M}_i\,(i = 1,\ldots,n)$ in p-dimensional Euclidean space $\mathbb{R}^p$, where $p \in \{2,3\}$. The clocks of the receivers are synchronized. Now m signals are created at arbitrary positions $\mathbf{S}_j\,(j = 1,\ldots,m) \in \mathbb{R}^p$ at unknown time points t_j. The signal wavefront propagates spherically from the signals' origins $\mathbf{S}_j$ with constant signal velocity c. The signals arrive at the receivers at time points T_{ij}, which can be measured.

We assume that the signals are discrete, such that we can distinguish them by their time points. Besides, we assume that we can identify and filter echoes from surrounding walls and from obstacles, such that the receivers obtain the signal in direct line of sight.

Now the problem is to calculate the positions of the receivers $\mathbf{M}_i$, the positions of the signal origins $\mathbf{S}_j$, and the times t_j when the signals were created – only from the times T_{ij} when the signals arrived.

The mathematical constraints between the receivers and signals are described by the signal propagation equation

$$c\,(T_{ij} - t_j) = \|\mathbf{M}_i - \mathbf{S}_j\| \;, \tag{1}$$

where only the arrival times T_{ij} are known. All signal times t_j, receivers $\mathbf{M}_i$, and signal origins $\mathbf{S}_j$ are unknown in our setting. The signal velocity c is also given. $\|\cdot\|$ denotes the vector norm in Euclidean space.

An equation system is formed by the equations for n receivers and m signals. Depending on these numbers the equation system may be under-determined, uniquely determined, or over-determined, as we will discuss in the next section.

For the cases of three receivers in the plane, and four receivers in three-dimensional space, and under the assumption that the signals originate from a distance, the problem can be solved in closed form. Also, for a minimum number of eight receivers in the plane, respectively ten receivers in 3D space, the equation system can be solved directly [28].

To solve the equation system in general we have to square the equations [29]. When we distribute the equations we get squared and mixed terms. According to [11] and [28] it does not seem likely that efficient solutions to the problem in general can be found.

Non-linear approaches can solve the problem in many cases [10]. However, the iterative methods tend to run inevitably into local minima from which they cannot recover, even with repeated attempts. In this contribution we quantify the chance of running into local minima and we present an iterative method, the Cone Alignment algorithm, which increases the probability of successful solving.

1.3 *Solvability of the equation system*

Before we describe our solutions we discuss the degrees of freedom and the theoretical bounds on how many receivers n and signal origins m are necessary to find a unique solution. The minimal solutions have also been appealed to in [29].

We start the discussion for the two-dimensional case. Since the locations of all receivers and origins are unknown we face $2n+2m$ variables. Furthermore, we do not know when a signal has been created which adds m variables. Since we have no anchor points the number of variables reduces by two variables for translation (e.g. setting one node as origin) and one variable for rotation (e.g. setting another node on the x-axis and a third one with positive y-value).

We assume that all m signals are received at all n receivers which results in the following equation for the degrees of freedom $\mathcal{G}_2$ presented by the problem size:

$$\mathcal{G}_2(n,m) \;:=\; 2n + 3m - nm - 3 \tag{2}$$

If $\mathcal{G}_2(n,m) > 0$ then there is no unique solution for the problem, i.e. it is under-determined. There is a chance of a unique solution if $\mathcal{G}_2(n,m)$ equals zero. For negative values the problem is over-determined, which might allow the compensation of inaccuracies. See Table 1 for the two-dimensional case.

For the three-dimensional case the number of location variables is increased by $n+m$. Here, three variables can be set to a constant for the symmetry induced by translation and three variables for the rotation symmetry which leads to the following degrees of freedom, see Table 2:

$$\mathcal{G}_3(n,m) \;:=\; 3n + 4m - nm - 6 \tag{3}$$

Note that point and mirror symmetry is not covered by this discussion. Since we assume that there is abundant supply of ambient signals we can summarize that at least four receivers might allow the solution in the two-dimensional case when at least five signals are available. For the three-dimensional case of the problem five receivers for at least nine signals might be sufficient. However, in our simulations we saw that ambiguities remain which cannot be explained by symmetries. Stewénius [29] found 344 solutions to the problem of four receivers and five signals in the plane. In fact, six signal sources seem to be the minimum case for the problem.

signal	receivers											
sources	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	3	3	3	3	3	3	3	3	3	3	3	3
3	5	4	3	2	1	**0**	**-1**	**-2**	**-3**	**-4**	**-5**	**-6**
4	7	5	3	1	**-1**	**-3**	**-5**	**-7**	**-9**	**-11**	**-13**	**-15**
5	9	6	3	**0**	**-3**	**-6**	**-9**	**-12**	**-15**	**-18**	**-21**	**-24**
6	11	7	3	**-1**	**-5**	**-9**	**-13**	**-17**	**-21**	**-25**	**-29**	**-33**
7	13	8	3	**-2**	**-7**	**-12**	**-17**	**-22**	**-27**	**-32**	**-37**	**-42**
8	15	9	3	**-3**	**-9**	**-15**	**-21**	**-27**	**-33**	**-39**	**-45**	**-51**
9	17	10	3	**-4**	**-11**	**-18**	**-25**	**-32**	**-39**	**-46**	**-53**	**-60**
10	19	11	3	**-5**	**-13**	**-21**	**-29**	**-37**	**-45**	**-53**	**-61**	**-69**
11	21	12	3	**-6**	**-15**	**-24**	**-33**	**-42**	**-51**	**-60**	**-69**	**-78**
12	23	13	3	**-7**	**-17**	**-27**	**-37**	**-47**	**-57**	**-67**	**-77**	**-87**

Table 1. Degrees of freedom for the *two-dimensional* case. Non-positive values indicate potentially solvable problem instances.

signal	receivers											
sources	1	2	3	4	5	6	7	8	9	10	11	12
1	0	2	4	6	8	10	12	14	16	18	20	22
2	3	4	5	6	7	8	9	10	11	12	13	14
3	6	6	6	6	6	6	6	6	6	6	6	6
4	9	8	7	6	5	4	3	2	1	**0**	**-1**	**-2**
5	12	10	8	6	4	2	**0**	**-2**	**-4**	**-6**	**-8**	**-10**
6	15	12	9	6	3	**0**	**-3**	**-6**	**-9**	**-12**	**-15**	**-18**
7	18	14	10	6	2	**-2**	**-6**	**-10**	**-14**	**-18**	**-22**	**-26**
8	21	16	11	6	1	**-4**	**-9**	**-14**	**-19**	**-24**	**-29**	**-34**
9	24	18	12	6	**0**	**-6**	**-12**	**-18**	**-24**	**-30**	**-36**	**-42**
10	27	20	13	6	**-1**	**-8**	**-15**	**-22**	**-29**	**-36**	**-43**	**-50**
11	30	22	14	6	**-2**	**-10**	**-18**	**-26**	**-34**	**-42**	**-50**	**-58**
12	33	24	15	6	**-3**	**-12**	**-21**	**-30**	**-39**	**-48**	**-57**	**-66**

Table 2. Degrees of freedom for the *three-dimensional* case. Non-positive values indicate potentially solvable problem instances.

1.4 *Gradient descent method*

A common approach to non-linear problems is an iterative non-linear least-squares fit using the gradient descent method. The approach has shortly been mentioned with regard to this problem in [28]. More specifically, it was used by Biswas and Thrun [10], who presented a solution based on likelihood maximization of the signal and receiver positions represented in a Bayesian network. As they assume the measurement uncertainties to be Gaussian distributed, the problem effectively reduces to minimization of the mean square error (cf. [10][1], Eq. (8))

$$\min_{\mathbf{M}_i,\mathbf{S}_j,t_j} \sum \left(t_j + \frac{1}{c}\|\mathbf{M}_i - \mathbf{S}_j\| - T_{ij} \right)^2 ,$$

which they optimize by using gradient descent. We have adapted this approach to compare to our proposed Iterative Cone Alignment algorithm.

In gradient descent a system of constraint equations is minimized, with a constraint for every pair of receiver and signal. We describe a constraint equation by

$$f_{ij} := c\,(T_{ij} - t_j) - \|\mathbf{M}_i - \mathbf{S}_j\| , \tag{4}$$

[1] Apparently, a minus sign is missing in Eq. (8) and in the previous Gaussian density function

We pursue to obtain solutions for $\mathbf{M}_i$, $\mathbf{S}_j$, and t_j by minimizing the objective

$$\underset{\mathbf{M}_i,\mathbf{S}_j,t_j}{\arg\min} \sum_{i=1}^{n}\sum_{j=1}^{m}(f_{ij})^2 \ . \tag{5}$$

The gradient descent method uses the first-order derivative of the equation system, the Jacobian matrix. In every iteration step k we calculate the Jacobian $\mathbf{Q}$ containing the partial derivatives for mn constraints and for $2n+3m$ unknowns in the planar case, respectively $3n+4m$ unknowns in three dimensions:

$$\mathbf{Q} := \begin{bmatrix} \frac{\partial f_{11}}{\partial \mathbf{S}_{1,1}} & \cdots & \frac{\partial f_{11}}{\partial t_m} & \frac{\partial f_{11}}{\partial \mathbf{M}_{1,1}} & \cdots & \frac{\partial f_{11}}{\partial \mathbf{M}_{n,p}} \\ \frac{\partial f_{12}}{\partial \mathbf{S}_{1,1}} & \cdots & \frac{\partial f_{12}}{\partial t_m} & \frac{\partial f_{12}}{\partial \mathbf{M}_{1,1}} & \cdots & \frac{\partial f_{12}}{\partial \mathbf{M}_{n,p}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{nm}}{\partial \mathbf{S}_{1,1}} & \cdots & \frac{\partial f_{nm}}{\partial t_m} & \frac{\partial f_{nm}}{\partial \mathbf{M}_{1,1}} & \cdots & \frac{\partial f_{nm}}{\partial \mathbf{M}_{n,p}} \end{bmatrix} .$$

$\mathbf{M}_{i,l}, \mathbf{S}_{j,l}$ denotes the l-th scalar of $\mathbf{M}_i, \mathbf{S}_j$. Furthermore, we combine the constraint functions into a vector $\mathbf{b} := (f_{11}, f_{12}, \ldots, f_{nm})^T$, containing the evaluation of mn function values. The values for $\mathbf{M}_i, \mathbf{S}_j$ are obtained from a state vector $\mathbf{u}^{(k)}$ of the form $\left(\mathbf{S}_{1,1}^T, \ldots, t_m, \mathbf{M}_{1,1}^T, \ldots, \mathbf{M}_{n,p}^T\right)^T$ in every iteration k. The operator $(\,\cdot\,)^T$ denotes the transposition.

According to (5) we write the quadratic error as $w = \left(\frac{1}{2}\mathbf{b}^T\mathbf{b}\right)$, which is proportional to the objective to minimize, and calculate the direction of the steepest *ascent*

$$\hat{\mathbf{u}} = \gamma\, \nabla w = \gamma\, \mathbf{Q}^T\mathbf{b} \ , \tag{6}$$

where γ is an adaptive factor for the step width.

In this way, we obtain an update vector $\hat{\mathbf{u}}$, where $-\hat{\mathbf{u}}$ is the direction of the steepest *descent*. The state vector is now updated by $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \hat{\mathbf{u}}$, where $\mathbf{u}^{(0)}$ has been initialized with appropriate initial estimates of signal and receiver positions. Here, we can only use randomized input. When the norm of the update vector is below a limit $\|\hat{\mathbf{u}}\| < \epsilon$ we abort the optimization.

For speedup we have added an extension to the optimization algorithm, and switch to the fast Gauss-Newton algorithm after some time, reducing the number of iterations when the gradient has become small during gradient descent optimization. The scheme of the Gauss-Newton algorithm is identical to the one of gradient descent, except for the calculation of the update. For the Gauss-Newton update we write the system of non-linear equations of the form (4) as a least squares equation system in matrix notation, and solve for $\hat{\mathbf{u}}$:

$$\mathbf{Q}^T\mathbf{Q}\hat{\mathbf{u}} = \mathbf{Q}^T\mathbf{b} \quad \Rightarrow \quad \hat{\mathbf{u}} = (\mathbf{Q}^T\mathbf{Q})^{-1}(\mathbf{Q}^T\mathbf{b}) \tag{7}$$

As the Gauss-Newton algorithm is very prone to divergence when applied to random initial positions, we do not use this method from the beginning. Instead, we rely on the robust gradient descent until the error function has become steady. However, when the gradient is shallow later during optimization, convergence in gradient descent tends to be very slow. Then, we switch to the fast the Gauss-Newton algorithm, making use of the quadratic convergence and reducing the number of iterations notably. As described later, we benefit from the Gauss-Newton algorithm not only after the gradient descent method, but also for finalization of our proposed Cone Alignment algorithm.

2 Iterative Cone Alignment

Many iterative approaches to the problem of TDOA use gradient descent, which is prone to local minima, or the Gauss-Newton method, which is additionally prone to divergence. Modifications of the gradient method with "momentum" require scenario-dependent adjustments of parameters.

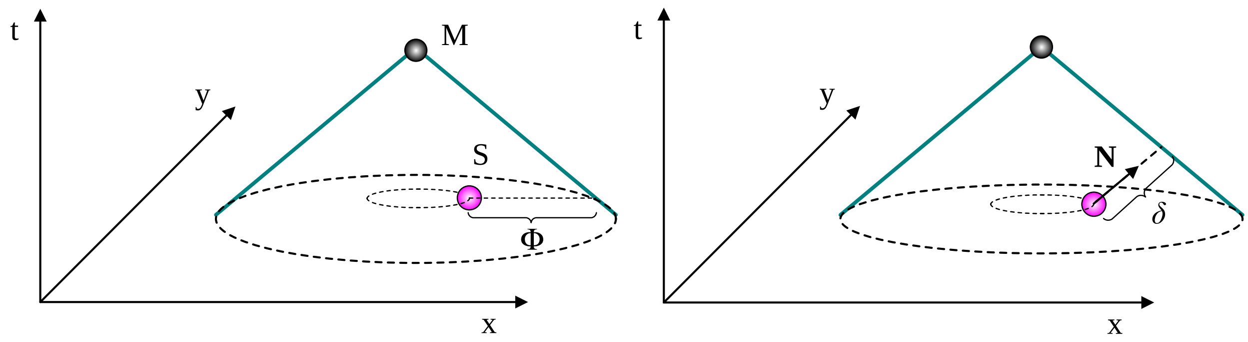


Figure 1. Cone representation of Eq. (8) for $p = 2$. *Left*: Signal source $\mathbf{S}$ resides offside the cone surface of receiver $\mathbf{M}$ and therefore it is not *valid* and $\Phi \neq 0$. *Right*: In order to restore validity $\mathbf{S}$ has to be moved in direction of $\mathbf{N}$.

We now present an iterative solution, our proposed Cone Alignment algorithm, which uses a geometric representation of the error function and a physical spring-mass simulation to optimize the problem for the general case. In the following we omit the indices i, j for clarity.

Consider a receiver $\mathbf{M}$ and a signal origin $\mathbf{S}$ in $p \in \{2, 3\}$-dimensional space. From the problem setting we know that

$$T = t + \frac{1}{c}\|\mathbf{M} - \mathbf{S}\| \ . \tag{8}$$

This equation describes a cone in the p+1-dimensional space where signal time t is added as an extra variable, see Fig. 1. The vector $(\mathbf{M}; T)$ is the apex of the cone. $(\mathbf{S}; t)$ describes a signal that occurred at position $\mathbf{S}$ at time point t. The expression $(\bullet\,;\,\circ)$ denotes a column vector of the form $(\bullet^T, \circ^T)^T$ where $(\bullet)$ is a column vector and $(\circ)$ is a scalar.

One can change both the position and time $(\mathbf{S}; t)$ of the signal source and the position vector $\mathbf{M}$ of the cone apex by moving it in space. According to Section 1.2, the receiver time T is fixed, so $(\mathbf{M}; T)$ is restricted to movements in p-dimensional space, whereas $(\mathbf{S}; t)$ can be moved in p+1 dimensions. If for all receivers $\mathbf{M}_1, \ldots, \mathbf{M}_n$, signal sources $\mathbf{S}_1, \ldots, \mathbf{S}_m$, and times $t_1, \ldots, t_m$ the Equations (8) are satisfied we receive a possible solution of the given problem. Of course, this does not necessarily imply we found the correct solution as the problem might be underconstrained. Recall that there is no absolute solution since we obtain only a relative localization.

2.1 *Error function*

We describe now an error function which corresponds to the potential energy of springs. Starting from an initial setting for all positions and time points our iterative approach greedily decreases the error function. We define the function in p+1-dimensional space as an expression of the violation of constraint (8) by

$$\Phi((\mathbf{D}; \tau)) := c\,\tau - \|\mathbf{D}\| \ , \tag{9}$$

where $\mathbf{D}$ is a difference vector in space and τ is the time of flight. $\Phi((\mathbf{M}; T) - (\mathbf{S}; t))$ describes the relation of a signal $\mathbf{S}$ emitted at time t relative to a receiver $\mathbf{M}$ which obtains the signal at time T. See Fig. 1, illustrating the Φ-function.

If Φ gives a non-zero value, we call this an *invalid* cone constraint. We require to manipulate the positions of $(\mathbf{S}; t)$ and $\mathbf{M}$ in space as described before in order to recover a valid position. We define the normal vector pointing towards the surface of the cone

$$\mathbf{N}_\ell := \left(\frac{\mathbf{S} - \mathbf{M}}{\|\mathbf{S} - \mathbf{M}\|};\ \frac{1}{c} \right) \ , \tag{10}$$

which we normalize to $\mathbf{N} := \frac{\mathbf{N}_\ell}{\|\mathbf{N}_\ell\|}$. The normalized direction vector $\mathbf{N}$ describes the shortest path from $\mathbf{S}$ to the cone surface of $\mathbf{M}$, cf. Fig. 1. Note that the dimension t is compressed by the signal velocity c. Scaling by c reveals that $\mathbf{N}$ is in fact orthogonal to the cone surface.

For the case that $t > T + \frac{1}{c}\|\mathbf{M} - \mathbf{S}\|$ and thus $\mathbf{N}$ does not intersect the cone, we choose $\mathbf{N} := (\vec{0}; -1)$ pointing along the time axis ensuring an intersection.

By construction there is a scalar $\delta \in \mathbb{R}$ such that $\Phi((\mathbf{M}; T) - (\mathbf{S}; t) + \delta\,\mathbf{N}) = 0$, i.e. an intersection point of $\mathbf{N}$ and the cone surface exists. δ equals the distance along $\mathbf{N}$ between $(\mathbf{S}; t)$ and the cone surface (Fig. 1). For every receiver i and signal j the distance can be computed by the intersect theorem

$$\delta_{ij} \;:=\; \frac{\Phi((\mathbf{M}_i; T_{ij}) - (\mathbf{S}_j; t_j))}{\Phi((\mathbf{M}_i; T_{ij}) - (\mathbf{S}_j; t_j)) - \Phi((\mathbf{M}_i; T_{ij}) - (\mathbf{S}_j; t_j) + \mathbf{N}_{ij})} \;. \tag{11}$$

For the goal to minimize the distances δ_{ij} between all signal senders and the cone surfaces of the receivers we require to manipulate their positions in $\mathbb{R}^{p+1}$. We aim to minimize the quadratic objective

$$E_{\text{sum}} \;:=\; \min_{\mathbf{M}_i, \mathbf{S}_j, t_j} \; \sum_{i=1}^{n}\sum_{j=1}^{m} (\delta_{ij})^2 \;,$$

which is proportional to the sum of the potential energy of springs [30]. By minimizing the potential energy of a physical mass-spring system representing the signal senders and receivers we minimize the objective E_{sum}. In case of a value $E_{\text{sum}} = 0$ we have found a scenario of senders and receivers, which is a possible explanation of the observed TDOA measurements.

2.2 *Spring-mass simulation*

We optimize the error objective E_{sum} by a simulation of a physical spring-mass system [30], calculating the signal source and receiver positions. The spring-mass system is based on *particles*, which are tuples $(\mathbf{x}^{[k]}, \mathbf{v}^{[k]}, m_0)$, representing the receivers and signals in p+1-dimensional space at discrete simulation times $k \in \mathbb{R}$. Again, we omit indices i and j in the following. The particles represent the physical properties position $\mathbf{x}$, velocity $\mathbf{v}$ and mass m_0. They obey Newton's law of inertia, i.e. velocity changes result only from the influence of forces. For every receiver tuple $(\mathbf{M}; T)$ and sender tuple $(\mathbf{S}; t)$ we create a particle.

For every sender particle we use the distance to the cone surface δ of a receiver particle to implement a force based on the spring equation $\mathbf{F} = -\kappa\,\delta\,\mathbf{N}$, where κ is a constant describing the spring stiffness. We apply the force $\mathbf{F}_{\text{spring}} = -\mathbf{F}$ to every receiver particle and the opposite force $\mathbf{F}_{\text{spring}} = \mathbf{F}$ to the corresponding signal particle, accelerating the sender particles towards the cone surface and the receiver particles in the opposite direction.

In cases, when movements are locked in a dimension, the respective component of the force vector is set to zero, thus preventing changes of the position in the respective dimension. In our case, this is used for the time component of the receivers. In more advanced cases, when restrictions of positions exist, for instance all receivers reside on a circle, penalty forces or local constraints [31] can be used.

In addition, we introduce quadratic damping, which is comparable to aerodynamic drag, stabilizing the simulation [30]:

$$\mathbf{F}^{[k]}_{\text{damp}} = -\lambda\,\|\mathbf{v}^{[k]}\|^2 \frac{\mathbf{v}^{[k]}}{\|\mathbf{v}^{[k]}\|} = -\lambda\,\|\mathbf{v}^{[k]}\|\;\mathbf{v}^{[k]} \;.$$

$\lambda > 0$ is a damping factor which we choose small to achieve high velocity of spring relaxation, but large enough to avoid the simulation to oscillate and become instable. For proper choice of a damping constant, damping reduces the velocity of a particle, and in this way its kinetic energy. The sum of forces for every particle is $\mathbf{F}_{\text{sum}} = \mathbf{F}_{\text{spring}} + \mathbf{F}_{\text{damp}}$.

The temporal integration is realized by a simple Euler-Cromer scheme with a timestep of h:

$$\mathbf{x}^{[k+h]} = \mathbf{x}^{[k]} + h\mathbf{v}^{[k+h]} \qquad\qquad \mathbf{v}^{[k+h]} = \mathbf{v}^{[k]} + \frac{h}{m_0}\mathbf{F}_{\text{sum}}^{[k]}$$

An Euler-Cromer scheme is *stable*, even in the undamped case, i.e. errors do not amplify over time. When using springs the scheme is *conditionally stable*, which means the stability is dependent on the parameters [32]. In practice the choice of spring and damping parameters is not an issue, as the range of stable operation is large.

We initialize the particles close to the origin in $p+1$-dimensional space, randomized by a small amount to avoid singularities. The initial signal source time is set to the minimum of all associated receiver timestamps. This is the closest guess we can do, as no positions are known a priori.

After the start, forces are calculated. Position and velocity updates are made accordingly to the Euler-Cromer scheme. The simulation runs until a termination condition has been met, where we obtain the positions $\mathbf{M}_i$ and the sender tuples $(\mathbf{S}_j; t_j)$ from the position vectors of the respective particles.

For the termination, either the overall energy function E_{sum} falls below a fixed threshold, or the overall particle velocity falls below a fixed limit, or a certain number of steps have been exceeded. If no TDOA error was presumed the latter two cases are an indication that the algorithm did not arrive at the zero of the error function. We call this a *local minimum*, an issue which is discussed later in Section 3.1.

3 Simulation

We have implemented the Cone Alignment algorithm in C++. Simulations were run in both, the two-dimensional and the three-dimensional case. For the signal velocity we choose the speed of sound at 20 °C, which is $c = 343\,\text{m/s}$.

For any number of receivers and signal sources $n, m \leq 14$ we created 100 random scenarios. Receivers and signal sources were placed in a two-dimensional, resp. three-dimensional space of 1000 meters edge length. For given randomly distributed signals in space we calculated the timestamps at every receiver. Then, the timestamp information was passed to our algorithm.

Since we have no anchor points we cannot directly compare the estimated positions to the reference positions (“ground truth”). We use singular value decomposition (SVD) to calculate a rotation and align the estimated positions with the reference positions [33]. After that we compare the output of the simulation to the ground truth positions by calculating the remaining root mean square error.

As an abort condition of the algorithm we chose an error threshold ϵ. In the successful case the remaining RMS error lay clearly below the threshold. If after 20.000 iteration steps the threshold could not be reached, the run was marked as not successful.

The algorithm converges after 1,000 to 10,000 iterations for $n, m \leq 14$ which gives an absolute runtime of 0.01 to 0.70 seconds on a standard 2 GHz PC, cf. Table 3. Although the runtime of one iteration step is in $\mathcal{O}(mn)$, as one spring equation is calculated for every pair of receiver and signal, the algorithm does not seem to follow this asymptotic time. Most interestingly, in our simulations the number of iterations decreases with increasing numbers of signal sources and receivers due to over-determination, i.e. a highly negative degree of freedom, as described in Section 1.3.

3.1 *Local minima*

In some cases the localization algorithm failed and got stuck in a local minimum of the error function. This opposes reconstruction errors due to under-determined scenarios, where constraints contain too little information and degrees of freedom remain. Local minima occured mainly in uniquely determined or over-determined scenarios.

The failure rate converges to zero with increasing number of signals, depicted in Fig. 2(a) for the two-dimensional case and in Fig. 2(b) for three dimensions. Comparing this observation with Table 1 and 2 shows that high failure rates correspond to small absolute degrees of freedom.

receivers	signals	iterations	duration (s)
2	2	1,815	0.013
14	2	4,359	0.132
2	14	5,086	0.148
5	5	9,087	0.108
7	7	5,620	0.170
10	10	4,636	0.333
14	14	4,259	0.680

Table 3. Average number of iterations and average experiment duration on a 2 GHz PC until the abort condition is satisfied in a 2D experiment. 100 runs were executed for every setting of receiver and signal numbers. The number of iterations decreases in the over-determined cases.

In a visual representation we saw that items were blocked on the wrong side of a line or a plane. We implemented an algorithm that mirrored them on the other side by way of trial. This successfully resolved local minima in some cases, but not in all. Obviously, some of the local minima are complicated and hardly comprehensible by geometrical considerations. For now, we disable the algorithm and instead use repetitions of Cone Alignment with different random initial positions. By this means, we could improve the rate of success notably, which is described in Section 3.3.

Furthermore, we ran experiments with simulated TDOA error. Here, the error in timestamping the signals at the receivers is assumed to be Gaussian distributed. Timing errors may be induced from synchronization errors and from imprecisions in determining the time points of signal events in the recorded audio. Errors of a standard deviation up to 200 ms were tested, which is a spatial equivalent of 70 m.

With increasing TDOA error both the average distance from the real positions and the tendency of local minima increased. We observed this tendency difficult to quantitize as with increasing error a local minimum is hard to distinguish from the global minimum in the least-squares sense, see Fig. 3 for an example.

For an evaluation of convergence to a solution for a given number of receivers and signals we calculated the remaining error after mapping the experimental positions to the corresponding ground truth positions using SVD. The experiments indicate, that for the two-dimensional case at least four receivers are necessary and sufficient. If the number of signals is fixed to three, deploying a sufficient number of receivers makes the localization error decrease to zero. In the three-dimensional case we observed similar results, with convergence for five receivers, respectively four signal sources. These observations correspond to our considerations from Section 1.3, where we predicted the reconstructability of all unknown positions for such numbers of signals and receivers.

We ran a direct comparison of the gradient method and the Cone Alignment. Both algorithms were run with and without the Gauss-Newton algorithm used afterwards. Again, we observe regions with higher failure rate for the gradient method, especially in the case of four receivers and in the case of three signal sources.

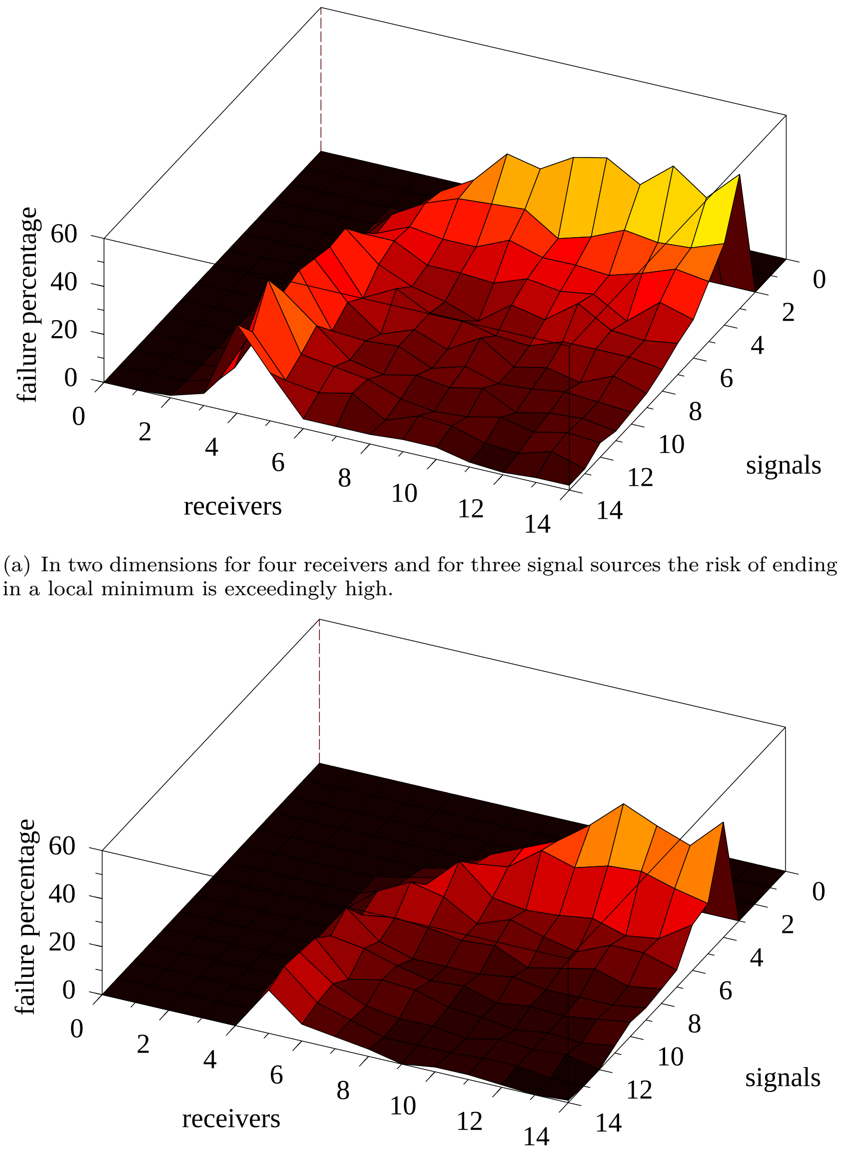
3.2 *Minimum case: Four receivers*

We focus on the case of four receivers in the plane, the smallest case in which positions can be calculated, which is especially interesting for application, as the receivers are usually the costly part. Our simulations indicate that the Cone Alignment algorithm has lower tendency to get stuck in local minima: Using Cone Alignment we achieve a lower failure rate for a varying numbers of signals. (Fig. 4). We observe the same with a fixed number of three signal sources.

As an explanation we suppose that the gradient descent method fails to escape local minima, as it can only decrease in its error function. In contrast, the particles of the Cone Alignment gather momentum while relaxing the spring constraints. In this way, barriers can be overcome towards a smaller minimum. As we implemented particle velocity as an imitation of physical springs we did not have to optimize a momentum parameter.

Furthermore, we observe a exceedingly high tendency to get stuck if we used the Gauss-Newton method after the gradient descent when we have four to six signal sources. We could not finally elaborate the reason for that.

Both algorithms, the Cone Alignment and the gradient method, benefit from the combination with the Gauss-Newton algorithm. We observed that scenarios with very shallow gradients were marked as “unsolved”



(a) In two dimensions for four receivers and for three signal sources the risk of ending in a local minimum is exceedingly high.

(b) In three dimensions the risk of local minima is highest for five receivers and for four signal sources.

Figure 2. Distribution of local minima for two and for three dimensions for the Cone Alignment algorithm. The risk of ending in a local minimum culminates at the minimum cases and converges to zero in overdetermined scenarios.

when an error threshold could not be reached after a maximum number of iteration steps. In several cases the threshold could be met when the Gauss-Newton algorithm was executed afterwards. This happens especially in ill-conditioned scenarios, for instance two of the four receivers are close. In general, the number of iteration steps is immensely reduced for both algorithms when the convergence is finalized with a subsequent run of the Gauss-Newton algorithm. With increasing number of both receivers and signals the ratio of local minima decreases. Also, the disparity between both algorithms diminishes.

3.3 *Towards a 100 % solution*

We have extended our algorithm and increased the success rate of finding the global minimum by repeated executions of the Cone Alignment algorithm, see Fig. 5. The repeated attempts come with increased computational power for finding a solution, but the calculations can trivially be executed in parallel. In the minimum case of four receivers and six signal sources in the plane we achieve a success rate of 99.4 % after 100 repeats with randomized initialization. Only 0.6 % of all cases remain stuck and unsolvable.

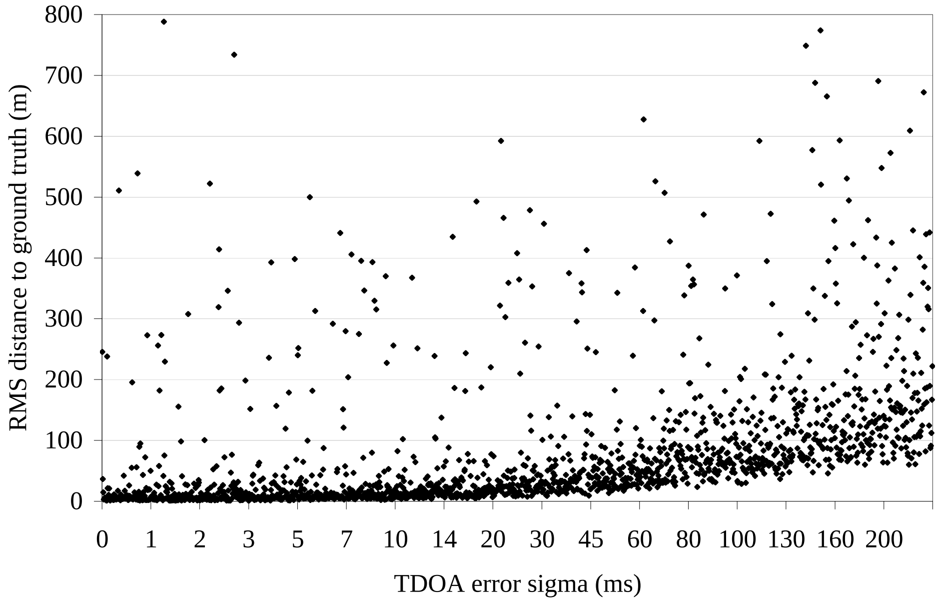


Figure 3. TDOA error experiment of the Cone Alignment algorithm for seven receivers and seven signal sources. For TDOA error steps from 0 to 200 ms a total of 1700 experiments were run. With increasing error the average distance from the real positions increases and local minima are harder to distinguish from the optimal solution.

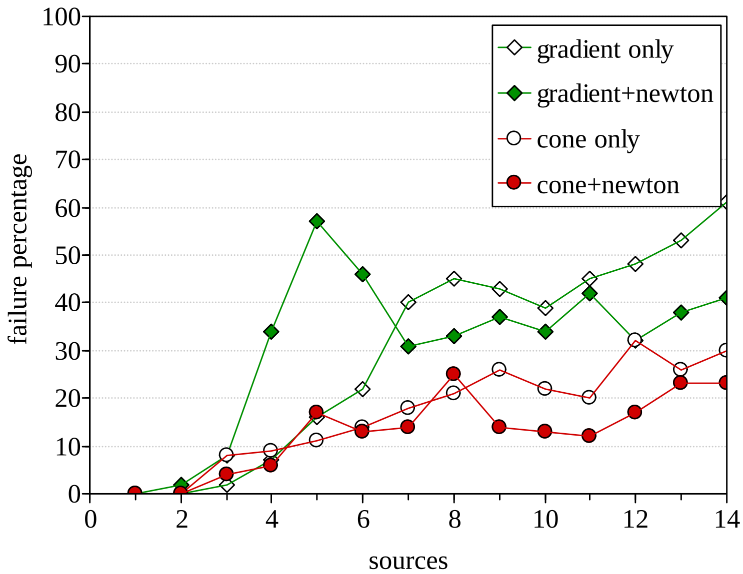


Figure 4. Comparison of failure ratio for 100 runs per combination. In the important case of 4 receivers in the plane the Cone Alignment (red marks) can find the global minimum more frequently than the gradient method (green marks).

In the case of the gradient descent method and the Gauss-Newton method combined we could not achieve such a high success rate. After 100 repeats still 2.4 % of all scenarios fail to be solved, which is more by a factor of four.

A least squares fit of the distribution in Fig. 5 appears to follow the power law $y = 1 - ax^b$. Omitting the first two data points we yield an exponent $b_{\text{cone}} = -1.02$ for the Cone Alignment and $b_{\text{grad}} = -0.83$ for the gradient descent method. The coefficient of determination is $R^2 > 0.98$ in both cases, suggesting the regression is trustable. The larger negative exponent b_{cone} is a clear indication that the Cone Alignment converges faster towards one and thus towards finding a solution for a scenario after a number of repeats.

As we can split larger scenarios into subsets of this size and merge them after solving a subset, we can solve larger scenarios in the same way. This form of repeating should work also for the other minimum cases, for 5 / 4 and for 7 / 3 receivers and signal sources, which we both found to be unique, and for the three-dimensional case.

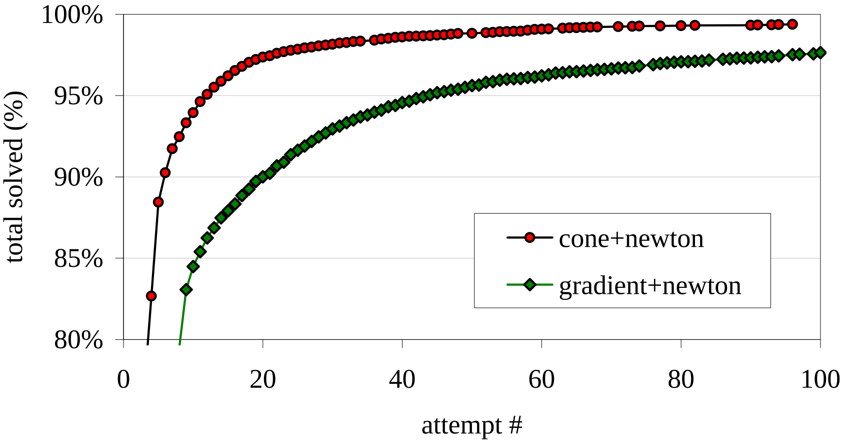


Figure 5. In the important minimum case of 4 receivers and 6 signal sources in 2D space the Cone Alignment algorithm solves 99.4% of all scenarios. With 12 attempts we solve more than 95 %. Using gradient descent we achieve only 97.6 % after 100 attempts.

4 Real-world experiments

We have verified our approach in real-world indoor and outdoor scenarios. In our experiments we use laptop computers and Apple iPhones with our software installed as receiver devices in a wireless network. Once our software is started the devices connect in a peer-to-peer network model and synchronize their clocks. Every device begins to record audio signals, either audible sound or ultrasound. In the case of audible sound we use the built-in microphones of the devices. Ultrasound signals are received by the laptops with external receiver devices attached, which we have built. From the discrete audio signals the time points of arrival are calculated using the synchronized time. The time points are committed to every participating device and the position calculations are executed locally.

TDOA localization with unsynchronized devices might be possible in general. For example distance estimation approaches might be used without synchronization if the offset between receivers is estimated from the average of the TDOA. However, the number of required sound events will increase to compensate for the additional variables. Another problem in unsynchronized localization is the drift of clocks which needs to be included into the mathematical model or eliminated by very precise calibration of the clocks.

Our algorithm relies on precise synchronization between the receivers. First, the connected clients negotiate one master device which acts as a time reference. Then, the other clients adjust their clocks to the reference. The calculation is done in an adaption of the Network Time Protocol algorithm. Both, the time offset and the timer drift is considered. With a 802.11 b/g Wi-Fi connection we achieve a synchronization precision of better than 0.1 ms. See [11] for details on the implementation and [34] for a summary of synchronization in wireless sensor networks.

Although the Cone Alignment algorithm is feasible in three-dimensional space, as described in Section 2, the following experiments will be conducted in a planar setting. This is a restriction for the sake of simplicity of the experimental setup. In a three-dimensional settings, the number of receivers would be higher, because of the determination of the equation system and the limited aperture of the microphones. Also, creating reference positions and sound signals is harder. Based upon the results of our simulations we expect similar results, if we manage to achieve TDOA measurements of the same precision as in two dimensions.

4.1 *Self-localization by clapping*

Our first real-world test took place in an outdoor setting on a green area on our campus. We arranged a scenario of four laptops and four Apple iPhones in a roughly elliptic formation of the dimensions 30 m × 30 m. The devices were connected over a dedicated Wi-Fi access point. Alternatively, one of the laptops could have been used as a Wi-Fi hotspot, making the setup independent from external infrastructures. The goal was to locate the positions of all laptops and iPhones just from natural sounds from the environment, as well as the positions of the sounds.

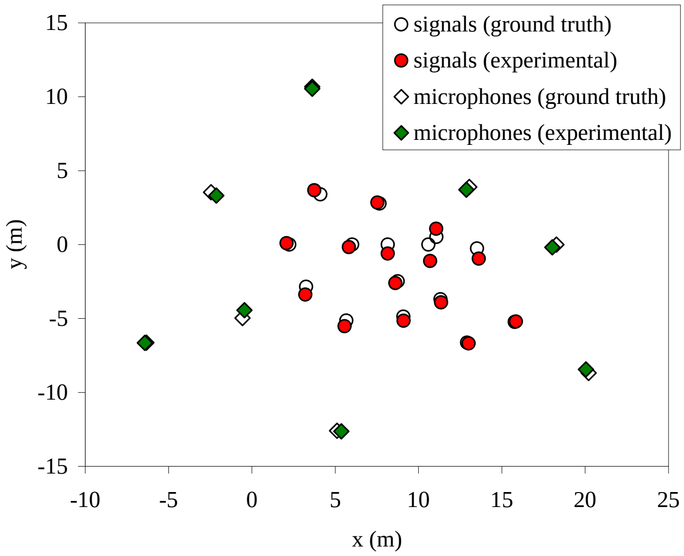


Figure 6. Four iPhones and four laptops to be located from 15 unknown sound signals. The experimenter claps two wooden planks while the assistant notes down the signal position. Two of the iPhones are on the wooden chairs in the background. The average location error of the receivers is 0.28 m ($\sigma = 0.14$ m)

With the network connection the timestamping software running on each device could communicate with the other instances and provide synchronization among all devices. With their built-in microphones they recorded any incoming sound event. The time points of the sounds were calculated by analysis of the audio stream. Sharp sound events like clapping, coughing, or finger clicking are detected by comparing the audio signal to an environment noise dependent threshold.

We charted the positions of all laptops and smartphones by measuring the distances to two anchor points using a measuring tape. The anchors were chosen as reference points for a Cartesian coordinate system. Then we calculated the x/y-coordinates of the devices up to a precision of 10 cm using trilateration (Fig. 6).

Now, an assistant was assigned to walk in the experiment field creating noises by clapping two wooden bars, which made a noise with sharp characteristics. The assistant was allowed to choose the locations of the sounds arbitrarily, but to move in between, such that the signals were well distributed.

The positions of the sound signals were marked with plastic cones on the floor. Since the assistant was free to move we could chart the sound signals only to a precision of 30 cm. Using the synchronous time base the software on every receiver calculated a synchronized timestamp of every sound event. A filter removed signals which had been missed by more than one computer, which occurred in some cases as a result of environmental noise. After filtering we identified a total of 15 sound signals at 15 noted positions.

Given these timestamps as input, the Cone Alignment algorithm computed the relative locations of the receivers and of the origins of the sounds. In the experiments we did not encounter the local minima issue as we had strictly over-determined scenarios.

The experimental data and the real-world positions were aligned by a congruent transformation by using singular value decomposition (SVD), minimizing the distances between experimental and ground truth positions. Recall, as pointed out in Section 1 and 2, that our approach does not use anchor points in space and provides only relative localization.

The average location error (euclidean distance) of the microphones after alignment was 0.28 m with a standard deviation of 0.14 m. The average error of the sound sources was measured to 0.39 m with a standard deviation of 0.28 m. The larger error of the sound sources might have been influenced by imprecise noise generation above the plastic cones and the measurement of their positions.

In these audio experiments we saw that in a controlled environment we could yield very precise timing of the audio events – and hearable sounds are cheaply available in many situations. An application of this may be locating the laptop computers in a computer pool, just by recording the noises of people in the room. By replacing the TDOA calculation by the more robust cross correlation [12], this would also work for human voices.

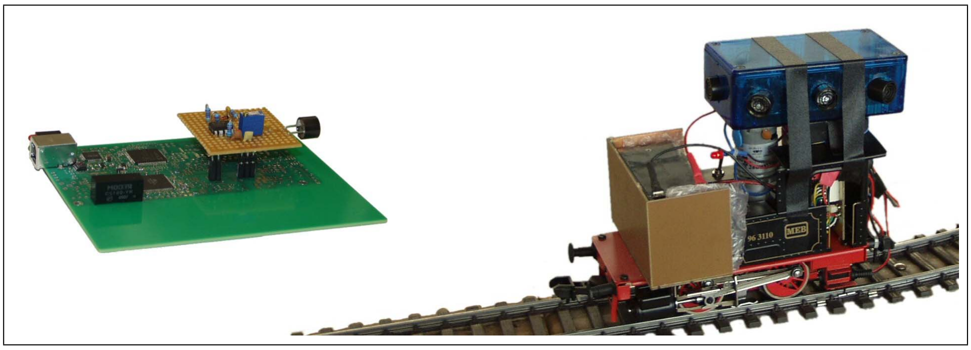


Figure 7. *Left*: Receiver platine with ultrasound capsule and USB connector. *Right*: Ultrasound beacon (blue rectangular box) with eight ultrasound capsules facing in all directions, attached to a model train.

For now, we presume that no additional clicking noise is created during the experiments. Otherwise the association of timestamps to sound events (the single claps of the experimenter) will become ambiguous. In the next section we present an ultrasound tracking system which is less vulnerable to association ambiguities.

4.2 *Configuration-free ultrasound tracking system*

Several tracking systems and approaches are available that achieve high precision in indoor and outdoor environments. Many of them are optical systems. However, most commercially available systems are expensive and need to be calibrated. Approaches using TDOA multilateration usually require receivers with calibrated locations, for which the positions have to be tediously measured by hand. This can be disadvantageous for industrial applications as these have to be easy to use.

We propose a novel tracking system for moving targets using our algorithm. It can quickly be set up, without the need to calibrate the positions of the devices. Of course, when the positions of at least three of the devices are given, the relative coordinates that we obtain can be converted to absolute coordinates.

We understand that audible sounds are not appropriate here, they would simply be annoying. We use ultrasound as a medium. Our ultrasound tracking system consists of a sender beacon and receivers that record and process the signals from the beacon. It has been assembled from off-the-shelf components and underprices most commercially available tracking systems.

The ultrasound beacon creates short ultrasound pulses at periodic intervals. With eight ultrasound capsules facing in all directions it creates an approximately isotropic signal. It can be carried by a person or it is attached to a moving unit, for example a model car or a model aircraft. This enables us to track a moving vehicle in real-time. As it is battery powered, it can be used independently from line voltage.

The interval of the ultrasound pulses can be freely chosen. It should be so large that signals arriving at the receivers can be distinguished. For example, in an experimental setup with the dimensions of 20 m the interval should be larger than 50 ms. In our experiment we use an interval of 300 ms.

We record the signal using external receiver devices with ultrasound microphones (Fig. 7). After filtering with an analog band-pass filter the signal is amplified and digitized by an analog-to-digital converter. Over a serial connection the data is forwarded to a processing computer, for example a laptop. Here, the data stream is searched for signal peaks, as in the case of audio signals. We send very short peaks of 1 ms with no information encoded in the signal.

Model train. In an experiment we track a moving model train, where the ultrasound beacon is attached to its roof (Fig. 7). On a very simple trajectory, a rectangle of the dimensions $1.8\,\mathrm{m} \times 3.9\,\mathrm{m}$ and a curve radius of 0.6 m, the train moves with a velocity of about 0.5 m/s (Fig. 8).

Five receivers are placed roughly in an oval around the track, at a distance of 4 – 7 m. As we conduct an experiment in the plane, we place the receivers at the same height as the beacon.

Next, the ultrasound microphones are roughly oriented towards the oval track and connected to adjacent laptops. With our software running they find each other in a Wi-Fi network and synchronize their clocks. Using a measuring tape we measured the positions of the ultrasound capsules of the receivers up to a precision of 3 cm. For the dimensions of the train track we describe the geometrical shape of the track.

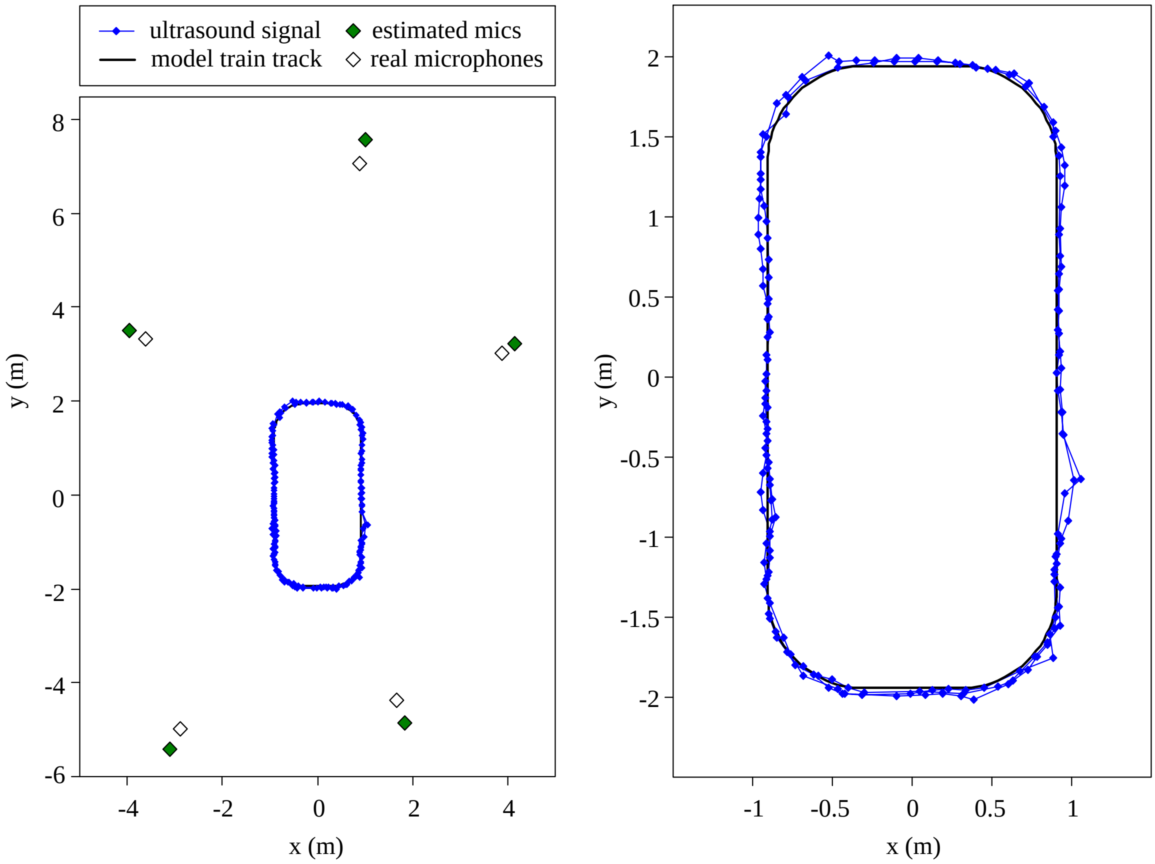


Figure 8. *Left*: Overview of the model train experiment with five receivers around an oval track of the dimensions 1.8 m × 3.9 m. The average error of the estimated receiver positions is 44.5 cm (σ = 7.7 cm). *Right*: Detailed view of the trajectory of the model train. For the estimated signal beacon position we observe a RMS track error of 2.5 cm.

For the tracking experiment we assume that the signals are spatially coherent, such that the moving beacon has limited velocity. In this way, we filter implausible timestamps. The phenomenon of multipath propagation, i.e. echoes from walls, was handled by issuing a dead time of the timestamp detector after every received signal. In rare cases a detector did not receive a signal in the direct path, but only a delayed echo of the same signal, which is then detected as the first signal. These false signals are detected and filtered in the way described.

After approximately three rounds the Cone Alignment algorithm got the TDOA data as the only input. Some signals were received by only four of the five receivers. These incomplete signals influence only those receivers by which they were received. We calculate both the unknown ultrasound receiver positions and the trajectory of the train on the track. The computed receiver positions were fit to the measured coordinates using SVD. Comparing the data, we find it well matching the ground truth. However, we observe some overestimation. The receivers show an average deviation from the real positions of 44.5 cm (σ = 7.7 cm). The overestimation is weakly pronounced for the trajectory of the model train. We observe only a small overestimation which results in a root mean square (RMS) track error of 2.5 cm.

In [35] Sippel et al. use a similar setup of an oval trajectory with a model train. In the indoor radar experiments they obtain an overall standard deviation of 3.6 cm, with notable overestimation of the real track of about 20 cm and with large outliers in case of disturbances. Using a laser scanner precise results are obtained, however the authors describe that the scanner is susceptible to losing track of the train. Both techniques require calibration and they are prone to influences of the environment. In contrast, our ultrasound system is not affected by obstacles in the environment, as long as a line-of-sight to the beacon exists, and the financial effort should be way below the costs of the radar and the laser system.

RC model car. For a second experiment we have redesigned the ultrasound beacon, as the quality of the results in the last experiment may have been affected by the rectangular shape of the box (cf. Fig. 7, right).

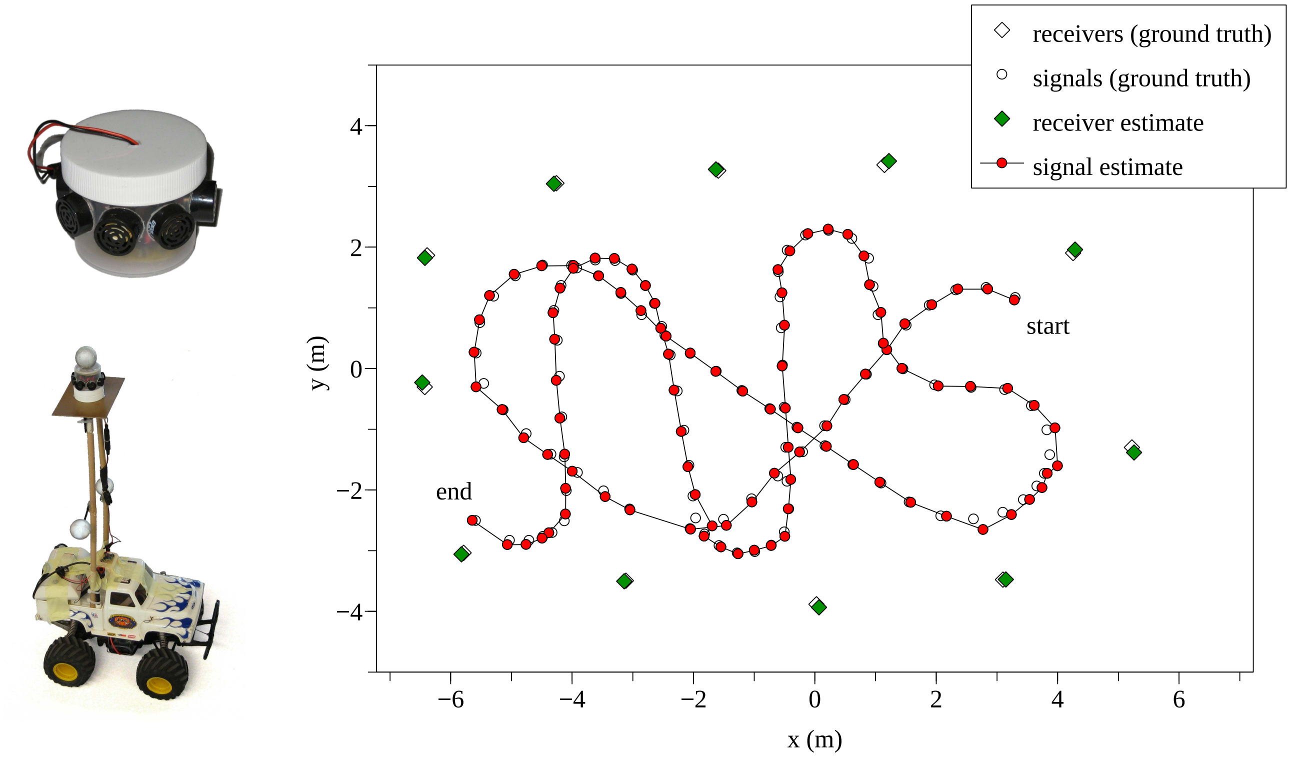


Figure 9. *Left*: The redesigned cylindrical ultrasound beacon, mounted on top of the RC model car. The reflective sphere is the marker for the optical reference system. *Right*: Trajectory of the RC car in the time interval 70 – 102 seconds. On its path, the model car generates ultrasound pulses every 0.3 seconds, which are received by up to eleven receivers.

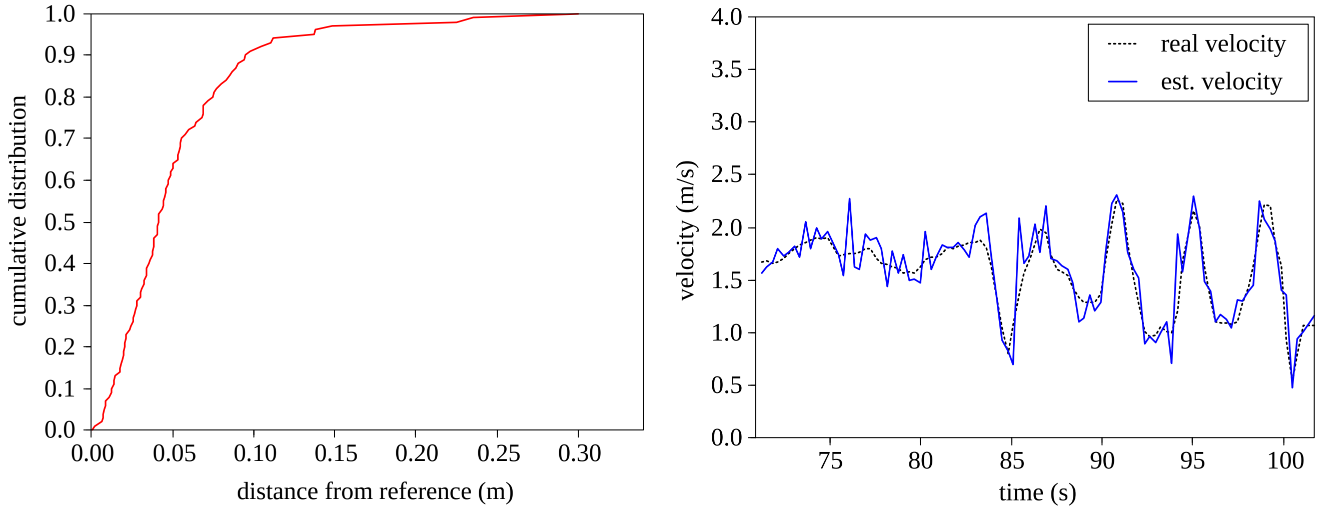


Figure 10. Evaluation of the time interval 70 – 102 seconds, see the overview in Fig. 9. *Left*: Cumulative distribution function (CDF) of the errors. The locations of the ultrasound beacon are estimated up to a mean error of 5.4 cm ($\sigma = 4.7$ cm), compared to the reference locations. *Right*: Estimated and reference velocity of the signal beacon.

Signals from the outer edge (i.e. the shorter edge) of the box will do a “head start”, compared to the long edge, resulting in a timing error of the receivers. With the new cylindrical casing the signal is now emitted from the same radius from the center to all directions in the plane (Fig. 9, left).

We attached the redesigned ultrasound beacon to a flagpole on top of a RC model car, and arranged the receivers in a range of $12\,\mathrm{m} \times 8\,\mathrm{m}$ (Fig. 9). This time, our focus was to create a large-scale experiment with signals received by only a subset of the receivers and to allow an arbitrary trajectory of the signal beacon. The model car was moving on the field between the receivers for several minutes in a random trajectory, generating a few hundreds of signals. Because of the limited range and aperture of the ultrasound capsules, and because the model car was guided through the whole experiment field, even outside the perimeter of the

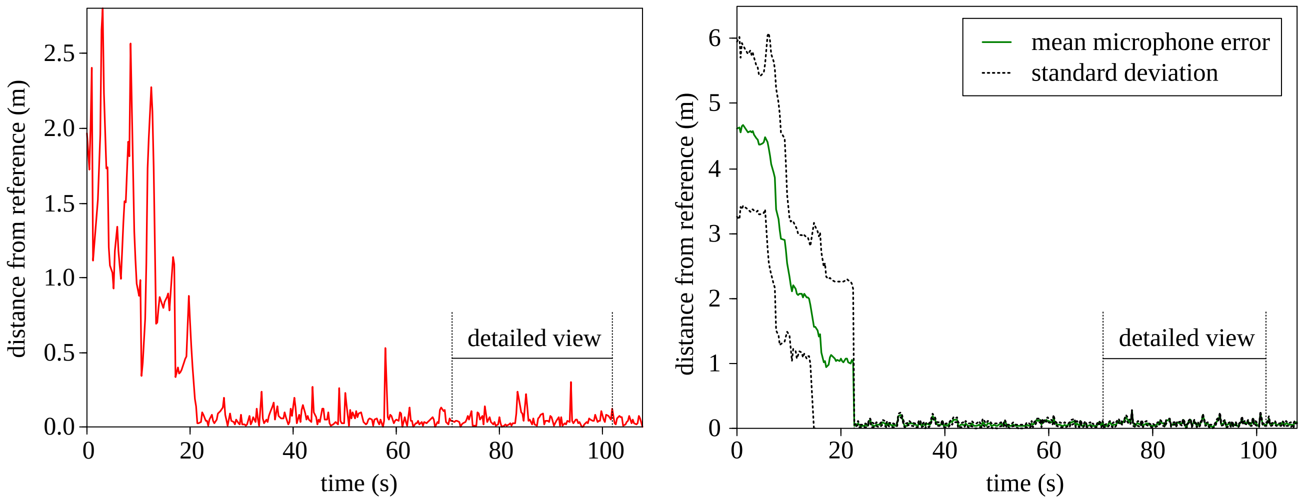


Figure 11. Iterative adding of the most recent signal, up to 80 signals. In the beginning neither the receivers nor the signal beacon positions are known. After 22 seconds a good estimation of the receivers is found, so the model car can be tracked. *Left*: Error of the ultrasound beacon. *Right*: Mean error and standard deviation of the receiver positions. *Detailed view* is the data segment in Fig. 9.

eleven receivers, the signals were not necessarily received by all of the receivers. We filtered the signals that were received by less than three receivers, as the positions of these signals cannot be calculated correctly, as the equation system is under-determined.

To generate references of the moving signal beacon we used a MotionAnalysis optical motion capture system (MoCap) with nine Raptor-E cameras. A reflective marker was attached above the ultrasound beacon which was tracked at a rate of 100 Hz, generating position references with a precision of millimeters (Fig. 9). In the same way we attached a marker exactly above the ultrasound capsule of the stationary receivers.

After the experiment a subset of the signals with the model car driving on a path throughout the experiment field was picked and filtered. We chose a section of the model car path with constant velocity of 1.8 m/s in the beginning, followed by variations of the velocity between 0.7 m/s and 2.3 m/s, which is a data set that we also used in [36]. The subset consisted of 101 signals, where the number of received timestamps per signal ranged from three to eleven. The data was evaluated by the Cone Alignment algorithm and the resulting position estimates for the ultrasound trajectory and for the receivers were mapped to the references.

The average error of the signal beacon in this part of the experiment is 5.4 cm ($\sigma = 4.7$ cm), the mean error of the microphones is 8.6 cm ($\sigma = 5.0$ cm). We did not observe the overestimation that occurred the previous experiment, most probably because of the redesigned beacon.

In an online evaluation of the experiment with recorded data, we iteratively added signals, up to a maximum of 80 signals, corresponding to about 24 seconds of time. Using the Cone Alignment algorithm we evaluate the positions of the most recent signal and of all receivers, given only the TDOA information. For this, we have created a modification of Cone Alignment, where particles are successively added into the running spring-mass simulation, while the abort condition of the algorithm is disabled. In this setting we, naturally, face an initialization problem of the receivers. No information is available in the beginning, neither the positions of the beacon, nor of the receivers. When the car starts to move, TDOA information is rare and not well distributed, most probably leading to defective position assumptions. As soon as the ultrasound signals arrive from different origins, the algorithm can recover, and the positions of signals and receivers can be correctly calculated.

In this experiment we evaluated a total of 350 signals from an interval of 110 seconds. 22 seconds after the beginning, a plausible estimation of the receivers was found, enabling precise tracking of the ultrasound beacon with an error of centimeters, see Fig. 11. We also saw in this experiment that Cone Alignment can run in real-time on a standard PC with moderate CPU load when new ultrasound signals from 11 receivers arrive every 300 ms.

In some rare cases we observed that the online algorithm got stuck in a local minimum, even when TDOA data was collected from throughout the experiment field. Re-initialization of the algorithm, as described in Section 3.3, may probably solve this problem.

5 Conclusions

We have addressed the problem of self-localization using nothing but TDOA information. Receivers have absolutely no knowledge about the signals and there is no assumption of the origin or the direction of the signals. There are no positional anchors among the receiver nodes. We only assume that the discrete signals can be distinguished from each other. The goal is to calculate the positions of the receivers and the positions of the signal origins – as well as the signal times implicitly.

In this contribution we have presented our novel Cone Alignment algorithm. The iterative spring-mass simulation solves the problem of relative localization in a energy minimization manner. Particles obeying Newton's law of inertia gather momentum while spring constraints are relaxed.

Like all iterative approaches to this problem the algorithm suffers from the risk of local minima. We have quantified the success rate of our algorithm and we have increased the probability of solving the scenario of four receivers and six signals to 99.4 %. Here, the algorithm outmatches the non-linear least squares approach, especially in the minimum case of four receivers in the plane.

In our real-world experiments we have proven the viability of our approach. We have located the positions of laptop and iPhone devices by natural sound signals from the surroundings. Furthermore, using our algorithm we have created an ultrasound localization system for real-time tracking of mobile objects, where precise indoor locations in the order of 10 centimeters were obtained. The tracking system does not require to measure the positions of reference receivers. As the only tasks we attach an ultrasound beacon to the moving vehicle and place the receivers at generally distributed, but arbitrarily chosen, positions in the room. Here, our tracking system can also cope with signals missed by some of the receivers.

If you are interested, please have a look at the video about our ultrasound tracking system, which is linked on our website: `http://cone.informatik.uni-freiburg.de`.

5.1 *Future work*

In graphical representations of the problem we have seen that we could solve the problem of local minima in some cases by flipping the particles. In this way, we might further increase the success rate of the algorithm.

We also plan to improve the practical aspects of our localization system. For many scenarios the assumption of discrete, distinguishable sound events is impractical. We envisage speaker tracking and locating ourselves by passing cars. This will require TDOA calculation by comparing audio signals using cross correlation. We expect this will extend the number of application scenarios for our technique.

For tracking experiments in three dimensions we plan to redesign our hardware prototypes. With the new devices we plan to conduct experiments in three dimensions using a model aircraft, such as a quadrotor.

Of great interest is also the question of unsynchronized localization. This would simplify the approach and would be helpful for unreliable network connections and for mobile networks, where synchronization is hard to achieve, as for example GSM or UMTS.

Acknowledgment

This work has partly been supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) within the Research Training Group 1103 (Embedded Microsystems).

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] 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.

[4] 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.

[5] G.C. Calafiore, L. Carlone, and M. Wei. A Distributed Gauss-Newton Approach for Range-based Localization of Multi Agent Formations. In *2010 IEEE International Symposium on Computer-Aided Control System Design (CACSD)*, pages 1152–1157. IEEE, 2010.

[6] W. Navidi, W.S. Murphy, and W. Hereman. Statistical methods in surveying by trilateration. *Computational statistics & data analysis*, 27(2):209–227, 1998.

[7] A. Efrat, D. Forrester, A. Iyer, S.G. Kobourov, C. Erten, and O. Kilic. Force-Directed Approaches to Sensor Localization. *ACM Transactions on Sensor Networks (TOSN)*, 7(3):1–25, 2010.

[8] K. Coogan, V. Khare, S.G. Kobourov, and B. Katz. MSDR-D Network Localization Algorithm. In *Proceedings of ALGOSENSORS '10*, 2010.

[9] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A Decentralized Network Coordinate System. In *Proceedings of the ACM SIGCOMM '04 Conference*, Aug. 2004.

[10] R. Biswas and S. Thrun. A Passive Approach to Sensor Network Localization. In *Proceedings of the International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). 2004 IEEE/RSJ*, volume 2, pages 1544–1549, 2004.

[11] T. Janson, C. Schindelhauer, and J. Wendeberg. Self-Localization Application for iPhone using only Ambient Sound Signals. In *Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 259–268, Nov. 2010.

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

[13] P. Pertila, M. Mieskolainen, and M.S. Hamalainen. Passive self-localization of microphones using ambient sounds. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1314–1318. IEEE, 2012.

[14] A. Savvides, C.C. Han, and M.B. Strivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] C. Hongyang, D. Ping, X. Yongjun, and L. Xiaowei. A Robust Location Algorithm with Biased Extended Kalman Filtering of TDOA Data for Wireless Sensor Networks. In *2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005. Proceedings.*, volume 2, pages 883–886. IEEE, 2005.

[21] R.L. Moses, D. Krishnamurthy, and R.M. Patterson. A Self-Localization Method for Wireless Sensor Networks. *EURASIP Journal on Advances in Signal Processing*, pages 348–358, 2003.

[22] S. Thrun. Affine Structure From Sound. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, Cambridge, MA, 2005. MIT Press.

[23] Yubin Kuang, Erik Ask, Simon Burgess, and Kalle Aström. Understanding TOA and TDOA Network Calibration using Far Field Approximation as Initial Estimate. In *International Conference on Pattern Recognition Applications and Methods*, 2012.

[24] Simon Burgess, Yubin Kuang, and Kalle Aström. Node Localization in Unsynchronized Time of Arrival Sensor Networks. In *Proceedings of the 21st International Conference on Pattern Recognition*, 2012.

[25] Johannes Wendeberg, Thomas Janson, and Christian Schindelhauer. Self-Localization based on Ambient Signals. *Theoretical Computer Science*, 453:98–109, 2012.

[26] R. Biswas and S. Thrun. A Distributed Approach to Passive Localization for Sensor Networks. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1248. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[27] C. Schindelhauer, Z. Lotker, and J. Wendeberg. Network Synchronization and Localization Based on Stolen Signals. In *Proceedings of 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2011.

[28] M. Pollefeys and D. Nister. Direct computation of sound and microphone locations from time-difference-of-

arrival data. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.*, pages 2445–2448. IEEE, 2008.

[29] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision.* PhD thesis, Lund University, April 2005.

[30] M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A Versatile and Robust Model for Geometrically Complex Deformable Solids. In *Computer Graphics International, 2004. Proceedings*, pages 312–319. IEEE, 2004.

[31] M. Gissler, M. Becker, and M. Teschner. Local Constraint Methods for Deformable Objects. In *Proc. of the 3rd Workshop in VR Interactions and Physical Simulation (VRIPHYS)*, pages 1–8, 2006.

[32] F. Vesely. *Computational Physics: An Introduction.* Springer, 2001.

[33] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 9(5):698–700, 1987.

[34] B. Sundararaman, U. Buy, and A.D. Kshemkalyani. Clock Synchronization for Wireless Sensor Networks: A Survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

[35] M. Sippel, W. Kuntz, and L. Reindl. New approach in precise laser tracking. In *IEEE Instrumentation and Measurement Technology Conference Proceedings*, pages 446–451, 2008.

[36] Johannes Wendeberg, Jörg Müller, Christian Schindelhauer, and Wolfram Burgard. Robust Tracking of a Mobile Beacon using Time Differences of Arrival with Simultaneous Calibration of Receiver Positions. In *Proceedings of 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012.