

# Anchor-free TDOA Self-Localization

Johannes Wendeberg\*, Fabian Höflinger†, Christian Schindelbauer\*, Leonard Reindl†

\**Department of Computer Science, University of Freiburg, Germany, {wendeber, schindel}@informatik.uni-freiburg.de*

†*IMTEK, University of Freiburg, Germany, {fabian.hoefflinger, reindl}@imtek.uni-freiburg.de*

**Abstract**—We present an approach for the localization of passive receiver nodes in a communication network. In our settings the positions of the nodes are unknown. 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 novel approach, the Iterative Cone Alignment algorithm, solves iteratively a non-linear optimization problem of time differences of arrival (TDOA) by a physical spring-mass simulation. Here, our algorithm shows a smaller tendency to get stuck in local minima than a non-linear least-squares approach.

The approach is tested in numerous simulations and in a real-world setting where 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 approach we estimate the trajectory of a moving model train with a precision in the range of centimeters.

## I. INTRODUCTION

The increasing availability and computational power of smartphones and handheld computers permits applications never before possible. Devices are equipped with high resolution displays, powerful processors, sensor systems, and microphones. However, the exact position of the phone remains subject to external infrastructures like the GPS system, GSM multilateration, or Wi-Fi based location services.

Localization approaches using these infrastructures heavily depend on the availability of the external systems. Infrastructures could 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 for many applications, 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. A sound source could be a finger snapping, coughing, or the tick sound of a metronome. 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.

978-1-4577-1804-5/11/\$26.00 © 2011 IEEE

## A. 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].

We focus on TDOA localization. TDOA data of audible sound can be obtained by discrete signal detection [10], [11] or by cross correlation of signals [12]. Ultrasound is used in [13], [14].

Usually, the receivers’ positions are known. Then, estimating a sender’s position using time differences of arrival can be addressed in closed form equations [15], [16] or by iterative approaches [17], [18]. Hongyang et al. use three anchor beacons [19]. Moses et al. use TDOA with additional angle information to locate unknown sender and receiver positions [20]. 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], [21], [22], [23]. 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. However, the correct solution cannot be found in every case. An upper error bound with signals in the unit disc is shown in [24]. A very elegant approach was proposed by Pollefeys and Nister [25]. The special case of ten or more microphones is solved in a linear approach without initialization and without assumptions on the positions.

## B. Problem setting

We address the problem of self-localization of receivers by using only TDOA information from unknown signal sources [11]. Consider a network of  $n$  receivers at unknown positions  $\mathbf{M}_i$  ( $i = 1, \dots, n$ ) in  $p$ -dimensional Euclidean space  $\mathbb{R}^p$ . The clocks of the receivers are synchronized. Now  $m$  signals are created at arbitrary positions  $\mathbf{S}_j$  ( $j = 1, \dots, m$ )  $\in \mathbb{R}^p$  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.

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

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

where  $\|\cdot\|$  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 fixed number of eight receivers in the plane, respectively ten receivers in 3D space, the equation system can be solved directly [25].

To solve the equation system in general we have to square the equations [26]. When we distribute the equations we get squared and mixed terms. According to [11] and [25] 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.

### C. Solvability

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 [26].

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:

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

signal sources	receivers											
	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 I

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

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 it equals zero. For negative values the problem is over-determined, which might allow the compensation of inaccuracies. See Table I 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 II:

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

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. Stevénius [26] 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.

### D. Gradient descent method

A common approach to non-linear problems is an iterative non-linear least-squares fit. The approach has shortly been mentioned with regard to this problem in [25].

Using gradient descent or Newton's method a system of constraint equations is minimized for every pair of receivers and signals. We describe the constraint equation by

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

In every iteration we pursue to minimize

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

signal sources	receivers											
	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 II

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

We state the non-linear equation (4) as a least squares equation system in matrix notation:

$$\mathbf{Q}^T \mathbf{Q} \mathbf{u} = \mathbf{Q}^T \mathbf{b} \quad (5)$$

The equation system consists of 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,

$$\mathbf{Q} := \begin{bmatrix} \frac{df_{11}}{d\mathbf{S}_{1,1}} & \dots & \frac{df_{11}}{dt_m} & \frac{df_{11}}{d\mathbf{M}_{1,1}} & \dots & \frac{df_{11}}{d\mathbf{M}_{n,p}} \\ \frac{df_{12}}{d\mathbf{S}_{1,1}} & \dots & \frac{df_{12}}{dt_m} & \frac{df_{12}}{d\mathbf{M}_{1,1}} & \dots & \frac{df_{12}}{d\mathbf{M}_{n,p}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{df_{nm}}{d\mathbf{S}_{1,1}} & \dots & \frac{df_{nm}}{dt_m} & \frac{df_{nm}}{d\mathbf{M}_{1,1}} & \dots & \frac{df_{nm}}{d\mathbf{M}_{n,p}} \end{bmatrix}$$

and of the function vector  $\mathbf{b} := (f_{11}, f_{12}, \dots, f_{nm})^T$  containing the evaluation of  $mn$  function values.  $\mathbf{M}_{i,l}, \mathbf{S}_{j,l}$  denotes the  $l$ -th scalar of  $\mathbf{M}_i, \mathbf{S}_j$ . Eq. (5) is solved for update vector  $\mathbf{u} := (\mathbf{S}_{1,1}, \dots, t_m, \mathbf{M}_{1,1}, \dots, \mathbf{M}_{n,p})^T$  in every iteration step  $k$ . The state vector  $\mathbf{u}^{(k)}$  is then updated by  $\mathbf{u}^{(k+1)} \leftarrow \mathbf{u}^{(k)} - \mathbf{u}$ , where  $\mathbf{u}^{(0)}$  has been initialized with appropriate values. Here, we can only use randomized input.

The equation system is usually solved by calculating the Inverse  $(\mathbf{Q}^T \mathbf{Q})^{-1}$ . However, we recommend using LU or QR decomposition for higher numerical stability.

We have observed that Newton's method is very prone to divergence when applied to arbitrary initial positions. Therefore, we use gradient descent and let it run for some time until the error function becomes steady before we switch to the fast Newton method. In particular, this led to higher probability of convergence.

## II. ITERATIVE CONE ALIGNMENT

For the general case no closed form solution is known so far. Iterative solutions of this problem use gradient descent, which is prone to local minima, or Newton's method, which is prone to divergence. Enhancements of the gradient method with "momentum" require scenario-dependent adjustments of parameters.

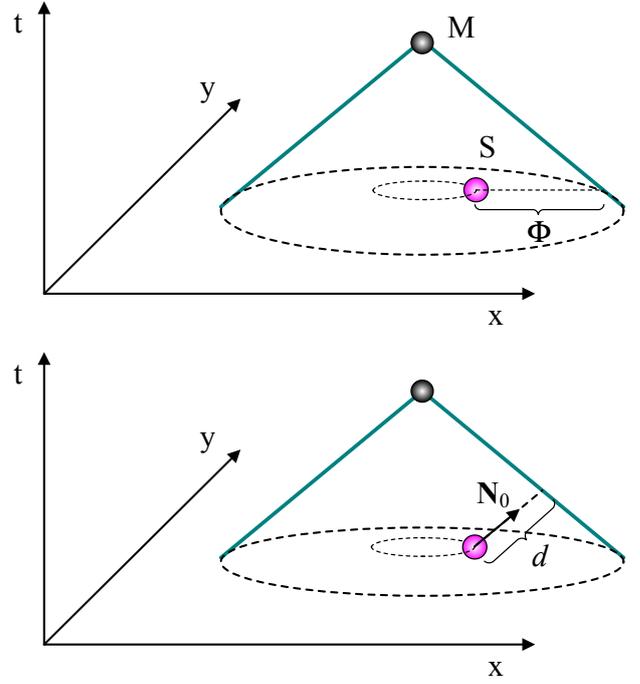


Fig. 1. Cone representation of Eq. (6) for  $p = 2$ . *Top*: Signal source  $\mathbf{S}$  resides offside the cone surface of receiver  $\mathbf{M}$  and therefore it is not *valid* and  $\Phi \neq 0$ . *Bottom*: Direction vector  $\mathbf{N}_0$  intersects the cone to restore validity.

We now present an iterative solution, the Cone Alignment algorithm, which uses a spring-mass simulation to solve 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

$$T = t + \frac{1}{c} \|\mathbf{M} - \mathbf{S}\| \quad (6)$$

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  $t$ . If for all receivers  $\mathbf{M}_1, \dots, \mathbf{M}_n$  and signal sources  $\mathbf{S}_1, \dots, \mathbf{S}_m$  these equations 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.

### A. Error function

We use an error function to describe 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:

$$\Phi((\mathbf{D}, t_{\mathbf{D}})) := ct - \|\mathbf{D}\| \quad (7)$$

If the error function  $\Phi$  gives a non-zero value, which we call an *invalid* location, one can change both the position and time  $(\mathbf{S}, t)$  of the signal source and the position vector  $\mathbf{M}$  of

the receiver by moving it in  $p+1$ -dimensional space in order to recover a valid position. Receiver time  $T$  is fixed by definition. We define:

$$\mathbf{N} := \left( \frac{\mathbf{S} - \mathbf{M}}{\|\mathbf{S} - \mathbf{M}\|}, \frac{1}{c} \right) \quad (8)$$

The normalized direction vector  $\mathbf{N}_0 := \frac{\mathbf{N}}{\|\mathbf{N}\|}$  describes the shortest path from  $\mathbf{S}$  to the cone surface of  $\mathbf{M}$  in respect of signal velocity  $c$ .

For the case that  $t > T + \frac{1}{c}\|\mathbf{M} - \mathbf{S}\|$  and thus  $\mathbf{N}_0$  does not intersect the cone, we choose  $\mathbf{N}_0 := (\vec{0}, -1)$  pointing along the time axis ensuring an intersection.

By construction there is a scalar  $d \in \mathbb{R}$  such that  $\Phi((\mathbf{M}, T) - (\mathbf{S}, t) + d\mathbf{N}_0) = 0$ .  $d$  equals the distance along  $\mathbf{N}_0$  between  $(\mathbf{S}, t)$  and the cone surface (Fig. 1). It can be computed by

$$d := \left( 1 - \frac{\Phi((\mathbf{M}, T) - (\mathbf{S}, t) + \mathbf{N}_0)}{\Phi((\mathbf{M}, T) - (\mathbf{S}, t))} \right)^{-1}. \quad (9)$$

We calculate a force to minimize  $d$  using the spring equation  $\mathbf{F} = -k d \mathbf{N}_0$  where  $k$  is a constant describing the spring stiffness. Applying  $\mathbf{F}$  to every receiver particle and  $-\mathbf{F}$  to the corresponding signal particle changes the locations and time points to minimize

$$E_{\text{sum}} = \sum_{i=1}^n \sum_{j=1}^m (d_{ij})^2$$

which is proportional to the sum of the potential energy of springs. In case of success all relations become valid which is the only way to yield a value  $E_{\text{sum}} = 0$ .

### B. Particle simulation

We compute the signal source and receiver positions by a simulation of a physical spring-mass system. It is based on *particles* which are tuples  $(\mathbf{x}_t, \mathbf{v}_t, m_0)$  representing the receivers and signals in  $p+1$ -dimensional space at discrete simulation times  $t$ . They have physical properties position  $\mathbf{x}$ , velocity  $\mathbf{v}$  and mass  $m_0$  obeying Newton's law of inertia. Velocity changes result from the influence of forces  $\mathbf{F}$ . In addition we introduce a quadratic damping, which is comparable to aerodynamic drag, stabilizing the simulation. The temporal integration is realized by a simple Euler-Cromer scheme with a timestep of  $h = 1$  ms:

$$\begin{aligned} \mathbf{x}_{t+h} &= \mathbf{x}_t + h\mathbf{v}_{t+h} \\ \mathbf{v}_{t+h} &= \mathbf{v}_t + \frac{h}{m_0}\mathbf{F}_t \end{aligned}$$

The simulation is initialized with all particles set to one spot in the  $p+1$ -dimensional space, jittered by randomization to avoid singularities. The initial signal source time is set to the minimum of all associated receiver timestamps. This is the closest position guess we can do by now, as no positions are given.

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. Then, either the overall energy function  $E_{\text{sum}}$  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.

### C. Evaluation of the algorithm

Since we have no anchor points we cannot directly compare our found positions to real-world positions ("ground truth"). As no positions are known to the algorithm, the final translation and rotation of the signal source and receiver network are not determined. For an evaluation of the quality of the algorithm we use singular value decomposition (SVD) to generate a rotation  $\mathbf{R}$  and align our found positions with the real-world positions [27].

Let  $G = \{\mathbf{g}\}$  and  $H = \{\mathbf{h}\}$  be a set of points in  $\mathbb{R}^p$  ( $p \in \{2, 3\}$ ), where  $G$  is the ground truth and  $H$  is our experimental data. We calculate the cross correlation  $\mathbf{W}$  by summing up the dyadic products of  $G$  and  $H$ . By subtracting the arithmetic centers  $\boldsymbol{\mu}_g$  and  $\boldsymbol{\mu}_h$  we eliminate the translation.

$$\mathbf{W} = \sum_{i=1}^{m+n} ((\mathbf{g}_i - \boldsymbol{\mu}_g)(\mathbf{h}_i - \boldsymbol{\mu}_h)^T)$$

Let the SVD of  $\mathbf{W}$  be

$$\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{U} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_p \end{bmatrix} \mathbf{V}^T$$

where  $\mathbf{D}$  is a diagonal matrix of singular values  $\sigma_i$  ( $1 \leq i \leq p$ ) of  $\mathbf{W}$ .  $\mathbf{U}$  and  $\mathbf{V}$  are unitary matrices.  $\mathbf{R} = \mathbf{U}\mathbf{V}^T$  creates a rotation  $\mathbf{R}$  with an optimal mapping of  $H$  to  $G$ :

$$H' = \{\mathbf{h}'\} = \{\mathbf{R}(\mathbf{h} - \boldsymbol{\mu}_h) + \boldsymbol{\mu}_g\} \approx \{\mathbf{g}\} = G$$

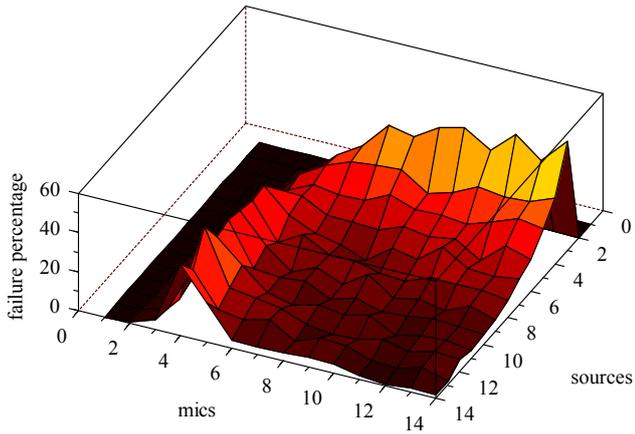
The remaining localization error is retrieved by calculating the root mean square (RMS) distance between  $G$  and  $H'$ .

## III. SIMULATION

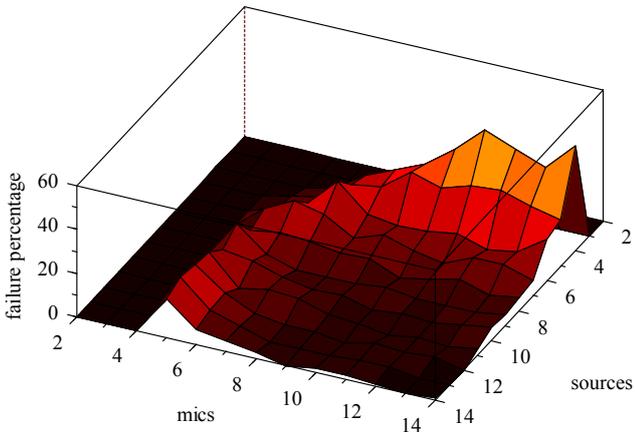
We have implemented the algorithm in C++. Simulations were run in both the two-dimensional and the three-dimensional case. For the signal velocity we choose the speed of sound at 20 °C, which is  $c = 343$  m/s.

The runtime of this algorithm is  $\mathcal{O}(mn)$  and it 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 PC. Most interestingly, the iteration count decreases with increasing numbers of sound sources and microphones due to over-determination (large negative degree of freedom).

For any number of microphones and sound sources  $n, m \leq 14$  we created 100 random scenarios. Microphones and sound sources were placed in a two-dimensional, resp. three-dimensional space of 1000 meters edge length. For given randomly distributed sound signals in space we calculated the



(a) In two dimensions for 4 receivers and for 3 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 5 receivers and for 4 signal sources.

Fig. 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.

timestamps at every microphone. Then, the timestamp information was given to our algorithm and finally we evaluated the quality of the result by applying SVD and comparing the output of the simulation to the ground truth positions. As an abort condition of the algorithm we chose an error threshold  $\epsilon$ . 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.

#### A. 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 occurred 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

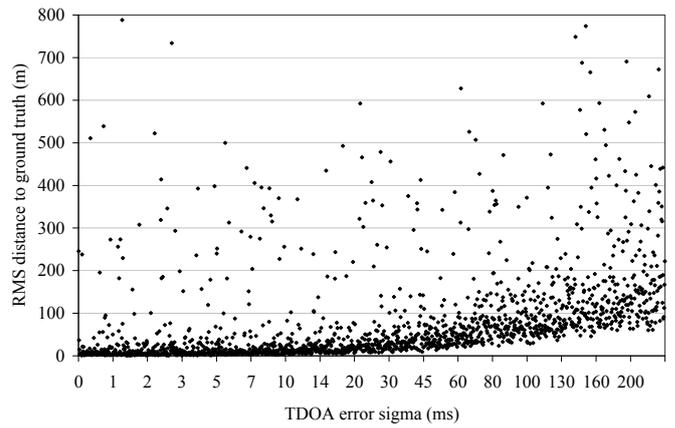


Fig. 3. TDOA error experiment of the Cone Alignment algorithm for 7 microphones and 7 signal sources. For TDOA error steps from 0 to 200 ms a total of 1700 experiments were run. With increasing error the average distance from the real positions increases and local minima are harder to distinguish from the optimal solution.

in Fig. 2(b) for three dimensions. Comparing this observation with Table I and II shows that high failure rates correspond to small absolute degrees of freedom.

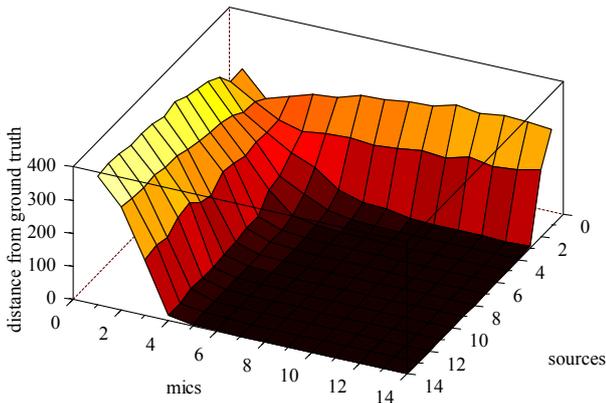
In a visual representation we saw that items were blocked on the wrong side of a line or a plane. We implemented an algorithm that mirrored them on the other side by way of trial. This successfully resolved local minima in some but not in all cases. Experiments with different initial positions had some improvement.

Furthermore, we ran experiments with simulated TDOA error. Here, the jitter in timestamping the signals at the receivers is assumed to be Gaussian distributed. In our experiments we found this to be realistic. The jitter may be induced from synchronization errors and from imprecisions in determining the timestamps. Errors of a standard deviation up to 200 ms were tested, which is a spatial equivalent of 70 m.

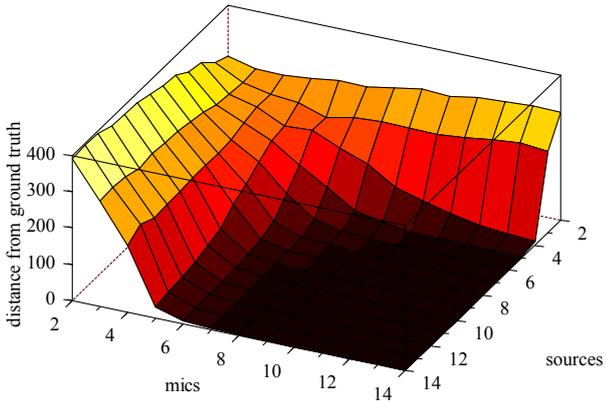
With increasing TDOA error both the average distance from the real positions and the tendency of local minima increased. We observed this tendency difficult to quantize as with increasing error a local minimum is hard to distinguish from the global minimum in the least-squares sense. See Fig. 3 for the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

For our experiments we calculated the remaining errors  $E_{rem}$  after mapping the experimentally calculated positions to the corresponding ground truth positions using SVD. We use the root mean square (RMS) distance as a metric.

For the two-dimensional case at least four microphones are necessary and sufficient to calculate the relative positions of all microphone and sound locations. For increasing numbers of sound sources the RMS distance error converges to zero. If the number of sound sources is fixed to three, deploying an increasing number of microphones makes the localization error decrease to zero (Fig. 4(a)). In the three-dimensional case we observed similar results, with convergence for five microphones, respectively four sound sources (Fig. 4(b)). These observations correspond to our considerations from



(a) Two-dimensional case. Given at least 3 sound sources, respectively 4 microphones, RMS position errors decrease.



(b) Three-dimensional case. Given at least 4 sound sources, respectively 5 microphones, RMS position errors decrease.

Fig. 4. Distance  $E_{\text{rem}}$  from real positions (“ground truth”) in a series of 100 random runs in two and in three dimensions for any combination of microphone and signal source numbers, filtered for successful runs.

Section I-C where we predicted the reconstructability of all unknown positions for such numbers of sound sources and microphones.

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

### B. Minimum case: Four receivers

We focus on the interesting case of four microphones in the plane, the smallest case in which positions can be calculated. 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 number of sources (Fig. 5). 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

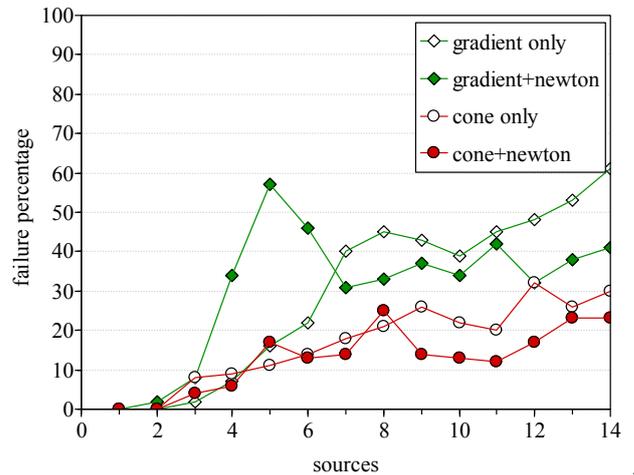


Fig. 5. Comparison of failure ratio for 100 runs per combination. In the important case of 4 microphones in the plane the Cone Alignment (red marks) can find the global minimum more frequently than the gradient method (green marks).

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 Newton’s method after the gradient descent when we have four to six sound sources. We could not finally elaborate the reason for that.

Both algorithms, the Cone Alignment and the gradient method, benefit from the combination with Newton’s method. We observed that scenarios with very shallow gradients were marked as “unsolved” when an error threshold could not be reached after a maximum number of iteration steps. In several cases the threshold could be met when Newton’s method was executed afterwards. In general, the number of iteration steps is immensely reduced for both algorithms when the convergence is finalized with Newton’s method.

With increasing number of both microphones and signals the ratio of local minima decreases. Also, the disparity between both algorithms diminishes.

### C. High solvability

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. 6. 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 microphones 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.

In the case of the gradient descent method and Newton’s 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.

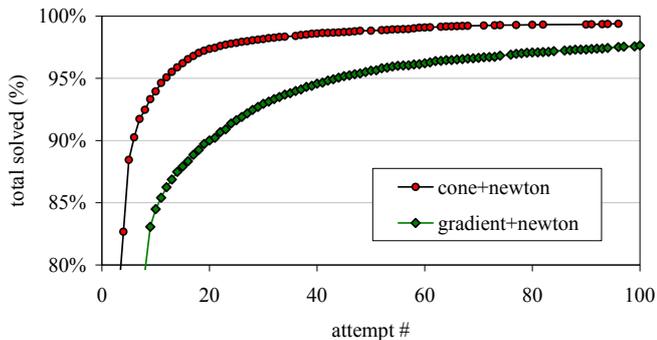


Fig. 6. In the important minimum case of 4 microphones and 6 signal sources in 2D space the Cone Alignment algorithm solves 99.4% of all scenarios. With 12 attempts we solve more than 95%. Using gradient descent we achieve only 97.6% after 100 attempts.

A least squares fit of the distribution in Fig. 6 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  $b_{\text{cone}}$  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 microphones and sound sources, which we both found to be unique, and for the three-dimensional case.

#### IV. 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 as receiver devices in a wireless network. Once our software is started the receivers 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 calculated from the mean 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 [28] for a summary of synchronization in wireless sensor networks.

##### A. 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  $\times$  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.

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.

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 caps on the floor. 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.

With these timestamps given 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 I, that our approach uses no anchor points in space and provides only relative localization.

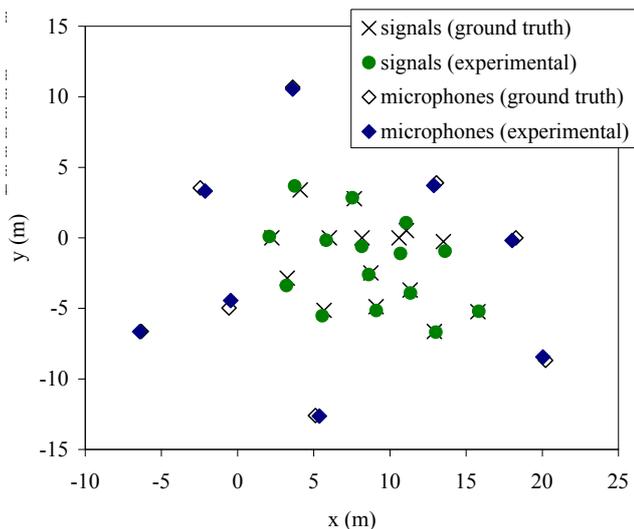


Fig. 7. Eight devices to be located by receiving 15 unknown sound signals produced by an assistant clapping two wooden bars. The average location error of the receivers is 0.28 m ( $\sigma = 0.14$  m).

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 higher error of the sound sources might be influenced by imprecise noise generation above the cones and the placement and measurement of the cone 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. However, we require 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 our next experiment we present an ultrasound beacon system which is less vulnerable to association ambiguities. The beacon can be attached to ground or airborne vehicles and provides a periodic signal. This enables us to track the moving vehicle in real-time.

### B. Ultrasound tracking system

A number of 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 require receivers with calibrated locations. Usually, the positions have to be tediously measured by hand. This can be disadvantageous for industrial applications as these have to be easy to use.

Now, we present a 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.

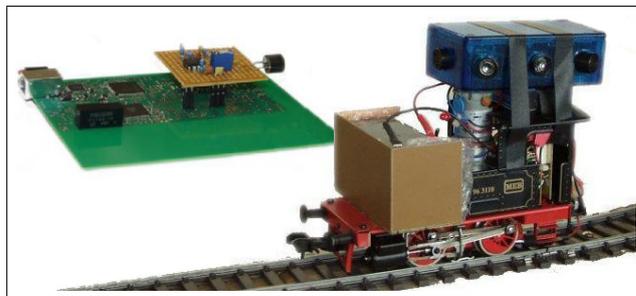


Fig. 8. *Left*: Receiver platine with ultrasound capsule and USB connector. *Right*: Beacon with eight ultrasound capsules facing in all directions, attached to a moving model train.

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 beacon creates short ultrasound pulses at periodic intervals. With eight ultrasound capsules facing in all directions it creates an approximately isotropic signal. For the lateral directivity of the beacon we have encountered an issue of noise cancelation at certain angles in the far field. This led to a spiky directivity diagram, but did not have noticeable impact on our experiments. The characteristic might be more homogeneous if the size of the beacon is smaller, because of the better overlap of the lobes of the sound capsules.

The interval of the ultrasound pulses can freely be 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.

Our ultrasound beacon can be carried by a person or it is attached to a moving unit, for example a model car or a model aircraft. It is battery powered so it can be used independently from line voltage.

We record the signal using external receiver devices with ultrasound microphones (Fig. 8). 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.

In an experiment we track a moving model train. On a very simple trajectory, an oval of the dimensions 3.9 m  $\times$  1.8 m, the train circles with a velocity of about 0.5 m/s. In our experiment the ultrasound beacon is attached to the roof of the model train (Fig. 8).

Five receivers are placed roughly in an oval around the track, at a distance of 4–6 m. As we conduct a 2D experiment we place the receivers at the same height as the beacon. Of course, our algorithm can be used in three-dimensional

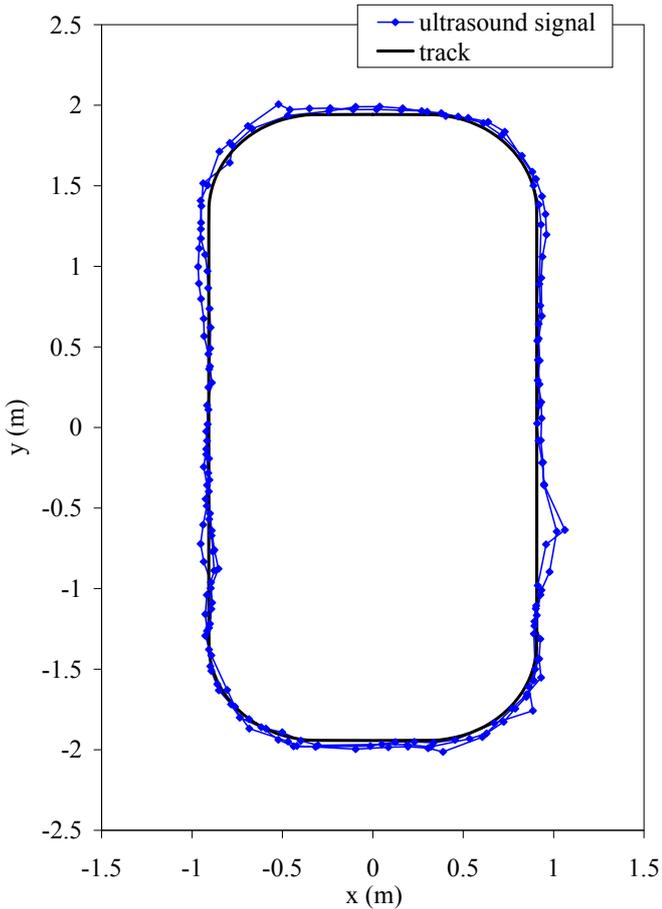


Fig. 9. Trajectory of the model train with the ultrasound beacon on top. We observe a small overestimation resulting in a RMS track error of 2.5 cm. The five receivers reside outside the track at a distance of 4–6 m.

settings, as described in Section II and III. Then, we distribute the receiver devices in space.

Next, the ultrasound microphones are roughly oriented towards the oval track and connected to adjacent laptops. With our software running they can find each other in a Wi-Fi network and synchronize their clocks.

Using a measuring tape we measured the positions of the ultrasound capsules up to a precision of 3 cm. For the dimensions of the train track we describe the geometrical shape of the track.

For the tracking experiment we assume that the signals are spatially coherent, in 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 encountered 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 can be detected and filtered in the way described.

After approximately three rounds the Cone Alignment algorithm got the TDOA data as the only input. 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 data. However, we observe some overestimation. The receivers show an average deviation from the real positions of 44.5 cm ( $\sigma = 7.7$  cm).

The overestimation is weakly pronounced for the trajectory of the model train (Fig. 9). We observe only a small overestimation which results in a root mean square (RMS) track error of 2.5 cm.

In comparison, the tracking system in [29] uses a similar setup of an oval trajectory with a model train. In the radar experiments an overall standard deviation of 3.6 cm is yielded, with notable overestimation of the real track and with large outliers in case of disturbances. Using a laser scanner precise results were 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.

## V. 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 microphones 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 a plane.

Our software platform establishes network communication between Windows notebooks and smartphones [11]. We require nothing but an active Wi-Fi connection in the same network. Then, we can synchronize the devices up to an order of 0.1 ms.

In our real-world experiments we have proven the viability of our approach. We have located the positions of laptop receivers with unknown audio signals from the surroundings.

Furthermore, using our algorithm we have created a quick-setup reference system for ultrasound vehicle tracking where

precise indoor locations in the order of centimeters are provided. There is no need to measure the positions of reference receivers. As the sole tasks we attach an ultrasound beacon to the moving vehicle and place the receivers at generally distributed, but arbitrarily chosen, positions in the room.

#### A. 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 scheme. For many scenarios the assumption of discrete, distinguishable sound events is impractical. We envisage speaker tracking and locating ourselves by passing cars. This requires to calculate TDOA 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 prototype. A smaller, spherical ultrasound beacon can be carried by a model aircraft, such as a quadrotor. Here, we will also address the directivity issue and increase the range and the reliability of the system.

Furthermore, we aim to improve the prediction of moving signals. Under the assumption of spatial coherence of signals we plan to apply filtering techniques like the Kalman filter or a Monte Carlo simulation. Then, we can estimate the beacon's position and interpolate in case of missing signals.

Of great interest is also the question of unsynchronized localization. This would simplify the approach immensely and would be helpful especially for unreliable network connections and for mobile networks like 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] C. Drane, M. Macnaughtan, and C. Scott, "Positioning GSM Telephones," *IEEE Communications Magazine*, vol. 36, pp. 46–54, 1998.
- [2] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, "Accurate GSM Indoor Localization," in *UbiComp*, 2005, pp. 141–158.
- [3] B. Ferris, D. Hähnel, and D. Fox, "Gaussian Processes for Signal Strength-Based Location Estimation," in *Proceedings of Robotics: Science and Systems Conference (RSS)*, 2006.
- [4] M. L. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," in *Proceedings of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems*, 2004, pp. 174–183.
- [5] G. 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)*. IEEE, 2010, pp. 1152–1157.
- [6] W. Navidi, W. Murphy, and W. Hereman, "Statistical methods in surveying by trilateration," *Computational statistics & data analysis*, vol. 27, no. 2, pp. 209–227, 1998.
- [7] A. Efrat, D. Forrester, A. Iyer, S. Kobourov, C. Erten, and O. Kilic, "Force-Directed Approaches to Sensor Localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 3, pp. 1–25, 2010.
- [8] K. Coogan, V. Khare, S. Kobourov, and B. Katz, "MSDR-D Network Localization Algorithm," in *Proceedings of ALGOSENSORS '10*, 2010.
- [9] F. Dabek, R. Cox, F. Kaashoek, and R. 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, vol. 2, 2004, pp. 1544–1549.
- [11] T. Janson, C. Schindelhauer, and J. Wendeborg, "Self-Localization Application for iPhone using only Ambient Sound Signals," in *Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Nov. 2010, pp. 259–268.
- [12] J.-M. Valin, F. Michaud, J. Rouat, and D. 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)*, 2003, pp. 1228–1233.
- [13] A. Savvides, C. Han, and M. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, pp. 166–179.
- [14] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 32–43.
- [15] L. 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*, vol. 57, pp. 4598–4615, Dec. 2009.
- [16] M. Gillette and H. Silverman, "A Linear Closed-Form Algorithm for Source Localization From Time-Differences of Arrival," *Signal Processing Letters, IEEE*, vol. 15, pp. 1–4, 2008.
- [17] D. J. Torrieri, "Statistical Theory of Passive Location Systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 2, pp. 183–198, March 1984.
- [18] D. Carevic, "Automatic Estimation of Multiple Target Positions and Velocities Using Passive TDOA Measurements of Transients," *IEEE Transactions on Signal Processing*, vol. 55, pp. 424–436, Feb. 2007.
- [19] 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.*, vol. 2. IEEE, 2005, pp. 883–886.
- [20] R. Moses, D. Krishnamurthy, and R. Patterson, "A Self-Localization Method for Wireless Sensor Networks," *EURASIP Journal on Advances in Signal Processing*, pp. 348–358, 2003.
- [21] T. Janson, C. Schindelhauer, and J. Wendeborg, "Self-localization Based on Ambient Signals," in *Algorithms for Sensor Systems*, ser. Lecture Notes in Computer Science. Springer, 2010, vol. 6451, pp. 176–188.
- [22] S. Thrun, "Affine Structure From Sound," in *Proceedings of Conference on Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2005.
- [23] R. Biswas and S. Thrun, "A Distributed Approach to Passive Localization for Sensor Networks," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1248.
- [24] C. Schindelhauer, Z. Lotker, and J. Wendeborg, "Network Synchronization and Localization Based on Stolen Signals," in *Proceedings of 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2011.
- [25] 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*. IEEE, 2008, pp. 2445–2448.
- [26] H. Stewénius, "Gröbner basis methods for minimal problems in computer vision," Ph.D. dissertation, Lund University, Apr. 2005.
- [27] K. Arun, T. Huang, and S. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 5, pp. 698–700, 1987.
- [28] B. Sundararaman, U. Buy, and A. Kshemkalyani, "Clock Synchronization for Wireless Sensor Networks: A Survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.
- [29] M. Sippel, W. Kuntz, and L. Reindl, "New approach in precise laser tracking," in *IEEE Instrumentation and Measurement Technology Conference Proceedings*, 2008, pp. 446–451.