Localization Solely based on Ambient Signals

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

February 2010

We present two approaches for the localization of passive nodes in a communication network using ambient radio or sound signals. In our settings the communication nodes have unknown positions. They are synchronized but do not emit signals for localization and exchange only the time points when environmental signals are received. The signals occur at unknown positions and times, but can be distinguished. Since no anchors are available, the goal is to determine the relative positions of all communication nodes and the environmental signals.

The first approach, Iterative Cone Alignment, deals with arbitrary positions of signals. It solves iteratively a non-linear optimization problem of time differences of arrival (TDOA) by a physical spring-mass simulation. Given a sufficient number of signals the approach converges towards a unique solution.

The second approach, Ellipsoid TDOA, demonstrates a closed form solution assuming the signals originate from far distances. The TDOA characterize an ellipse from which the distances and angles between three network nodes can be inferred.

Both approaches are tested in numerous simulations and in a real world setting where the relative positions of notebooks are determined utilizing only the sound produced by a girl with a glass bottle and a toy gun.

Contents

1 Introduction **3**

1.1 Motivation . . . 3

1.2 Related Work . . . 3

1.3 Problem setting . . . 4

1.4 Solvability . . . 5

2 Iterative Cone Alignment **7**

2.1 Error function . . . 7

2.2 Particle simulation . . . 9

2.3 Evaluation of the algorithm . . . 10

2.4 Simulation . . . 11

3 Ellipsoid TDOA method for distant sources **13**

3.1 TDOA ellipse . . . 15

3.2 Linear regression . . . 16

3.3 Three dimensional case . . . 17

3.4 Simulation . . . 19

3.5 Summary . . . 23

4 Real world experiments **26**

4.1 Time synchronisation . . . 27

4.2 Experiments . . . 27

4.2.1 Position reconstruction . . . 28

4.2.2 Ellipsoid TDOA method . . . 31

5 Conclusions **33**

5.1 Applications . . . 34

5.2 Future work . . . 35

References **35**

6 Appendix **38**

6.1 Reformulation as Optimization Problem with Polynomials of Degree 4 . . . 38

6.2 Newton approximation . . . 38

1 Introduction

The increasing mobility of computing devices, like smart phones, PDAs, laptops, and tablet computers is a motivation to revisit the localization problem from a fresh perspective.

1.1 Motivation

The usual approach is to include special hardware like GPS receivers, which adds extra monetary cost and power consumption. However, in shielded areas and for small distances such location hardware cannot solve the problem. This is in particular the case for sensor networks in houses or tunnels. Then, the standard approach is to use anchor points in the communication network and calculate the positions by the time of arrival (TOA), time difference of arrival (TDOA) or by the received signal strength indication (RSSI) of radio signals.

Our approach starts with the following idea. Suppose we have a number of devices with microphones in a room which are connected by a communication network, e.g. mobile phones or laptop computers. Now, somebody walks through the room snapping fingers. Solely based on the time when these sound signals are received, all distances and angles betweeen network nodes are computed. Or in a different scenario consider a sensor network installed by biologists to record bird tweeting in a forest. From the recorded tweeting it is possible to compute the relative positions of sensors, and when and where a bird has tweeted.

Of course an approach, that computes the positions of network nodes from random environmental sound signals, can also be used to calculate positions from received radio signals from TV satellites, closed WLANs, or signals from any other mobile devices. So, any ambient radio signal pollution enables localization.

1.2 Related Work

Localization with *known* receiver or sender positions is a broad and intensive research topic with a variety of approaches. A popular application is GSM localization of mobile phones. Various techniques exist, including angle/direction of arrival (AOA/DOA), time of arrival (TOA, "time of flight") and time difference of arrival (TDOA) [1]. U-TDOA is a provider-side GSM multilateration technique that needs at least three synchronized base stations. Denoising techniques improve accuracy [2]. As a client side implementation needs special hardware, it is hardly prevalent in common mobile phones. Instead, many approaches introduce a distance function based on the received signal strength indication (RSSI). Stable results in the range of meters can be achieved by fingerprinting using a map of base stations [3].

Similar is localization using the RSSI function of WiFi signals. Methods include Bayesian inference [4], semidefinite programming for convex constraint functions [5, 6] a combination of WiFi and ultra sound for TOA measurements like the Cricket system [7] or combinations of methods [8]. TOA distance functions can be used [9, 10]. WiFi Beamforming uses sensor arrays to determine signal directions [11, 12].

RSSI evaluation usually comes with difficulties for indoor localization due to the unpredictability of signal propagation [13]. We focus on TDOA analysis in our approach. For TDOA localization of sound and RF signals there is a basic scheme of four or more known sensors locating one signal source. This is solved in closed form [14, 15] or with iterative methods [16, 17, 18]. TDOA determination can be done by cross correlation of pairs of signals. An optimal shift between signals is calculated, corresponding to the angle of the signal [19, 20, 21]. However, we use signals with a characteristic peak.

Moses et al. [22] use DOA and TDOA information to solve the problem of unknown sender and receiver positions. Though sounding similar to our problem, both problem settings differ fundamentally. The additional DOA information enables the authors to apply some sort of "bootstrapping": Initial starting points can be found to solve the problem incrementally. However, the DOA acquisition requires expensive equipment like directed receivers or receiver arrays. To our knowledge our problem setting of unknown sender and receiver positions without any further information but TDOA has never been addressed so far.

1.3 Problem setting

Given a communication network of n synchronized nodes $\mathbf{M}_1, \dots, \mathbf{M}_n$, where $\mathbf{M}_i \in \mathbb{R}^p$ denotes the unknown position in two- or three-dimensional Euclidean space $\mathbb{R}^p$ ($p \in \{2, 3\}$). Now m sound (or radio) signals are produced at unknown time points $t_{\mathbf{S}_1}, \dots, t_{\mathbf{S}_m}$ and at unknown locations $\mathbf{S}_1, \dots, \mathbf{S}_m \in \mathbb{R}^p$. We assume that each signal can be received and identified at the receiver and we assume that it travels with a fixed speed c, which is the speed of sound or light depending on the signal type. So, each signal $\mathbf{S}_j$ arrives at receiver $\mathbf{M}_i$ at time $t_{\mathbf{M}_i,\mathbf{S}_j}$ which is the only input given in this problem setting. Since we assume synchronized clocks we can measure this time up to an error margin which we assume to be Gaussian distributed.

The problem is to compute all locations of receivers and signal origins, solely from the time when an environmental signal is received at the receivers. Of course then the signal times can be computed from this information. Although all our experimental work concentrates on sound and thus the speed of sound, all our findings also refer to electromagnetic waves or sound signals in other environments, e.g. water.

The mathematical constraints can be described using the signal velocity c, the time $t_{\mathbf{S}_j}$ of signal creation and the time $t_{\mathbf{M}_i,\mathbf{S}_j}$ when the signal is received at receiver $\mathbf{M}_i$:

$$c(t_{\mathbf{S}_j} - t_{\mathbf{M}_i,\mathbf{S}_j}) = |\mathbf{S}_j - \mathbf{M}_i|_2 \tag{1}$$

where

$$|\mathbf{S} - \mathbf{M}|_2 = \sqrt{(x_{\mathbf{S}} - x_{\mathbf{M}})^2 + (y_{\mathbf{S}} - y_{\mathbf{M}})^2 + (z_{\mathbf{S}} - z_{\mathbf{M}})^2}$$

denotes the Euclidean distance in three-dimensional space.

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

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

We will present two solutions for this problem. First we consider the general case and present the Iterative Cone Alignment, an iterative numeric solution based on an energy minimization approach. In our simulations and experiments the algorithm finds a solution if the number of received signals is large enough. Since one can assume that environmental signals are not a scarce resource, this algorithm leads to an accurate measurement of receivers and signals in real-world scenarios.

Then we move on to the case where the signal sources are so far from the receivers that the time difference at two receivers depends only on the angle between the signal beam and the line between the two receivers. We present the Ellipsoid TDOA method, an elegant closed form solution for the two- and three-dimensional case – for three receivers in two dimensions and four receivers in three dimensions. We extend the solution to the case of realistically distributed large numbers of signals.

Finally, we show how our algorithms perform in a real-world experiment. Here, a girl walks through a PC pool clanking a bottle and shooting a toy gun at some random locations, which is the sole information our software needs to compute the relative locations of the computers.

1.4 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.

We start the discussion for the three-dimensional case. Since the locations of all receivers and origins are unknown we face $3n + 3m$ 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 three variables for translation (e.g. setting one node as origin) and three variables for rotation (e.g. setting another node on the x-axis and a third one on the plane defined by x- and y-axis).

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**

Figure 1: Degrees of freedom for the three-dimensional case. Non-positive values indicate potentially solvable problem instances.

We assume that all m signals are received at all n receivers which results in the following equation for the degrees of freedom $\mathcal{G}_3$ presented by the problem size:

$$\mathcal{G}_3(n,m) = 3n + 4m - nm - 6 \tag{2}$$

If $\mathcal{G}_3(n,m) > 0$ then there is no unique solution for the problem, i.e. it is under-defined. There is a chance of a unique solution if it equals zero. For negative values the problem is over-defined, which might allow the compensation of inaccuracies, see Fig. 1.

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

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

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 five receivers might allow the solution in the three-dimensional case when at least nine signals are available. For the two-dimensional case of the problem four receivers for at least five signals might be sufficient.

Now we present an approximative solution, the Iterative Cone Alignment, which is based on an iterative approach.

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**

Figure 2: Degrees of freedom for the two-dimensional case. Non-positive values indicate potentially solvable problem instances.

2 Iterative Cone Alignment

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

$$t_{\mathbf{M},\mathbf{S}} = t_{\mathbf{S}} + \frac{|\mathbf{S} - \mathbf{M}|_2}{c} \tag{4}$$

This equation describes a cone in the p+1-dimensional space where time $t_{\mathbf{S}}$ is added as an extra variable, see Fig. 3. If for all receivers $\mathbf{M}_1, \ldots, \mathbf{M}_n$ and signal sources $\mathbf{S}_1, \ldots, \mathbf{S}_m$ these equations are satisfied we receive a valid solution of the given problem. Recall that there is no unique solution since we obtain only a relative localization.

2.1 Error function

Starting from an initial setting for all positions and time points we use an iterative approach which greedily decreases an error function. The error function is then used to minimize the potential energy of springs.

The difference vector $(\mathbf{D}, t_{\mathbf{D}}) = (\mathbf{S}, t_{\mathbf{S}}) - (\mathbf{M}, t_{\mathbf{S},\mathbf{M}})$ describes $(\mathbf{S}, t_{\mathbf{S}})$ in relation to $(\mathbf{M}, t_{\mathbf{M}})$. $(\mathbf{D}_{ij}, t_{\mathbf{D}_{ij}})$ denotes a certain pair of $(\mathbf{S}_j, t_{\mathbf{S}_j})$ and $(\mathbf{M}_i, t_{\mathbf{M}_i})$ $(1 \leq i \leq n, 1 \leq j \leq m)$. We define the error function:

$$\begin{aligned} \Phi((\mathbf{D}, t_{\mathbf{D}})) &:= c t_{\mathbf{D}} - |\mathbf{D}|_2 \\ &= c(t_{\mathbf{S}} - t_{\mathbf{M},\mathbf{S}}) - |\mathbf{S} - \mathbf{M}|_2 \end{aligned} \tag{5}$$

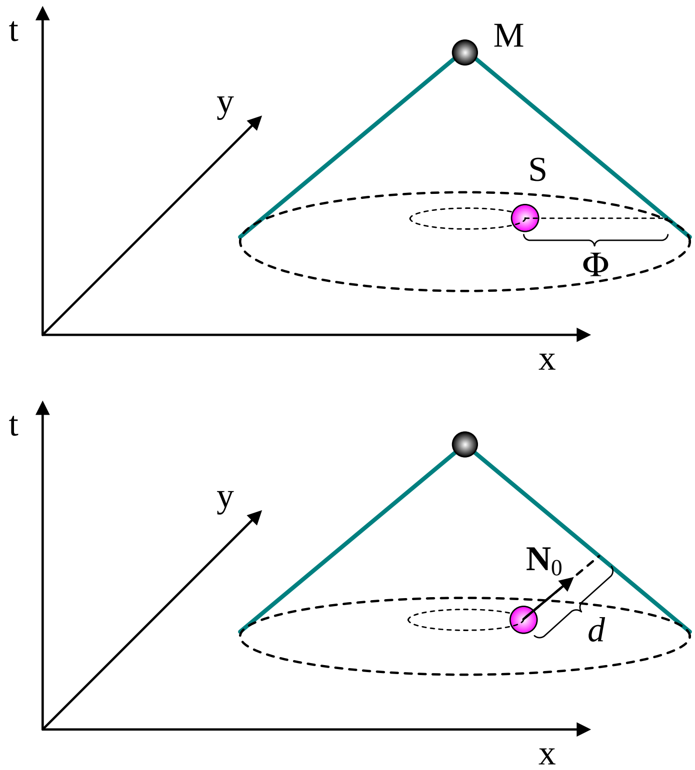

Figure 3: Cone representation of equation (4). *Top*: Signal source $\mathbf{S}$ resides offside the cone surface of receiver $\mathbf{M}$ and therefore it is not *valid* and $\Phi \neq 0$. *Bottom*: Direction vector $\mathbf{N}_0$ restores validity.

If the error function gives a non-zero value, which we call an *invalid* location, one can change both the position and time vector $(\mathbf{S}, t_\mathbf{S})$ 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 $t_{\mathbf{M},\mathbf{S}}$ is fixed by definition. We define:

$$\mathbf{N}_0 := \left(\frac{\mathbf{S} - \mathbf{M}}{|\mathbf{S} - \mathbf{M}|_2}, \frac{1}{c} \right) \tag{6}$$

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

For the case that $t_\mathbf{S} > t_\mathbf{M} + \frac{|\mathbf{S}-\mathbf{M}|_2}{c}$ 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 $\Phi((\mathbf{S}, t_\mathbf{S}) - (\mathbf{M}, t_{\mathbf{M},\mathbf{S}}) + d\mathbf{N}_0) = 0$ which can be computed by

$$d((\mathbf{D}, t_\mathbf{D})) = \frac{\Phi((\mathbf{D}, t_\mathbf{D}) + \mathbf{N}_0)}{\Phi((\mathbf{D}, t_\mathbf{D})) + \Phi((\mathbf{D}, t_\mathbf{D}) + \mathbf{N}_0)} \tag{7}$$

d is also the distance between $\mathbf{S}$ and the cone surface along $\mathbf{N}_0$. If receiver $\mathbf{M}_i$ missed out a signal $\mathbf{S}_j$ we set $d((\mathbf{D}_{ij}, t_{\mathbf{D}_{ij}})) = 0$.

Our iterative approach changes the locations and time points to minimize

$$E_\text{sum} = \sum_{i=1}^{n} \sum_{j=1}^{m} d((\mathbf{D}_{ij}, t_{\mathbf{D}_{ij}}))^2 \tag{8}$$

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

2.2 Particle simulation

We compute the source and receiver positions by a simulation based on a physical spring-mass system inspired by the spring relaxation method of Vivaldi [23]. It is based on *particles* representing a node in the $p+1$-dimensional space, which have physical properties of mass m_0 and velocity obeying the Newton's law of inertia. Velocity changes result from the influence of forces. In our setup this is achieved by springs, the tension of which is proportional to the error function Φ, along the direction $\mathbf{N}_0$. 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 $h = 1\,\text{ms}$:

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

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_{sum} falls below a fixed threshold or a certain number of steps has been exceeded. In the latter case no solution has been found and the approximation has failed.

2.3 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 the singular value decomposition (SVD) to generate a rotation $\mathbf{R}$ and align our found positions with the real world positions.

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.

$$\mathbf{W} = \sum_{i=1}^{m+n} \left((\mathbf{g}_i - \boldsymbol{\mu}_g)(\mathbf{h}_i - \boldsymbol{\mu}_h)^{\mathrm{T}} \right) \tag{9}$$

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

$$\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}} = \mathbf{U} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_p \end{bmatrix} \mathbf{V}^{\mathrm{T}} \tag{10}$$

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

$$H' = \{\mathbf{h}'\} = \{\mathbf{R}(\mathbf{h} - \boldsymbol{\mu}_h) + \boldsymbol{\mu}_g\} \approx \{\mathbf{g}\} = G \tag{11}$$

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

2.4 Simulation

We have implemented this algorithm in C++. For an intuitive graphical representation we implemented an interactive OpenGL frontend. It displays receiver and signal source items in a three-dimensional space and allows the user to rotate and zoom in and out. For the choice of parameters we concentrate on sound signals. All our findings also apply for radio signals.

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 and timestamps at the microphones were calculated. For given randomly distributed sound sources in space we calculated the timestamps at every microphone. Then, the timestamp information was given to our algorithm and finally we evaluated the quality of the result by applying the SVD.

Remaining errors were calculated by summing all root mean square (RMS) errors between each real and experimentally found object.

In our setup of 1000 m edge length a threshold of 100 m for the sum of all RMS distance errors can be set to tell successful from unsuccessful position reconstructions. In our simulations eventually the RMS distances for the successful case lie clearly below this threshold.

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 only in uniquely determined or over-determined scenarios. The failure rate converges to zero with increasing number of signals, depicted in Fig. 4 for the three dimensional case and in Fig. 5 for the two dimensional case. Comparing this observation with Table 1 and 2 shows that high failure rates correspond to small absolute degrees of freedom.

In a visual representation we saw items 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 no improvement. In the end we could not finally work out the conditions causing the issue which seems to be complex and requires further investigation.

The runtime of this algorithm is $\mathcal{O}(mn)$ and it converges after 2,000 to 9,000 iterations for non-failing instances for $n, m \leq 14$ which gives an absolute runtime of 0.01 to 0.68 seconds on a standard PC (Athlon XP 2600+). Most interestingly, the iteration count decreases to 4,000 iterations with increasing numbers of sound sources and microphones due to over-determination (large negative degree of freedom).

For the three-dimensional case at least five microphones are necessary and sufficient to calculate the relative positions of all microphone and sound locations. For increasing number of sound sources the approximation error (RMS distance sum) converges to zero. If the

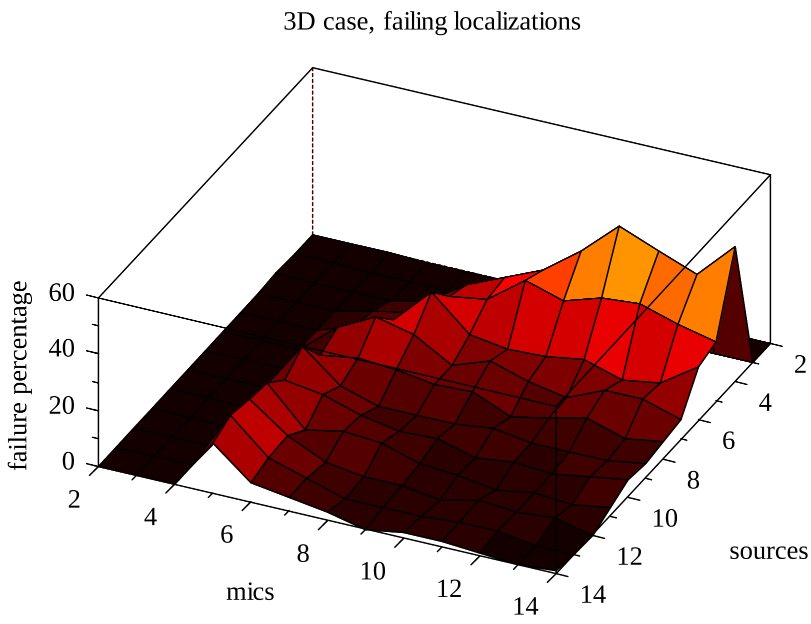

Figure 4: Failing runs in percent for three dimensions.

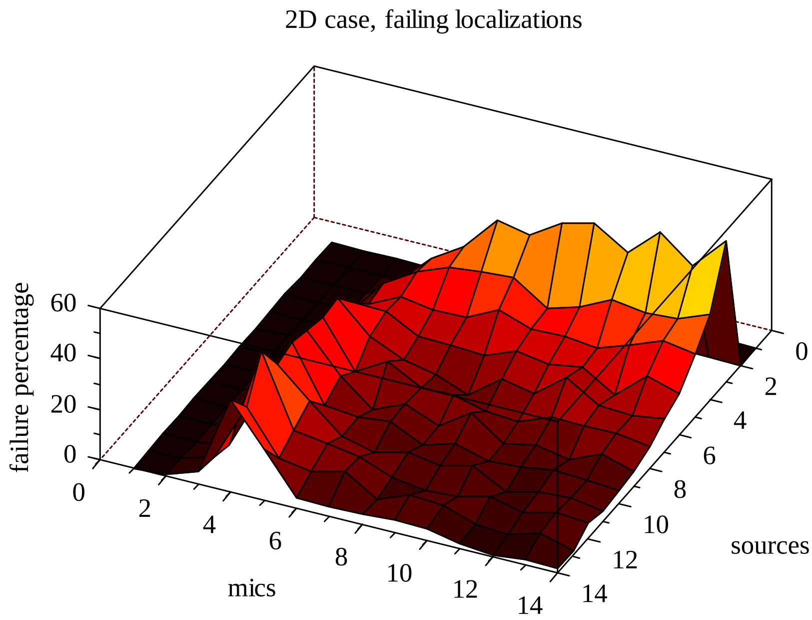

Figure 5: Failing runs in percent for two dimensions.

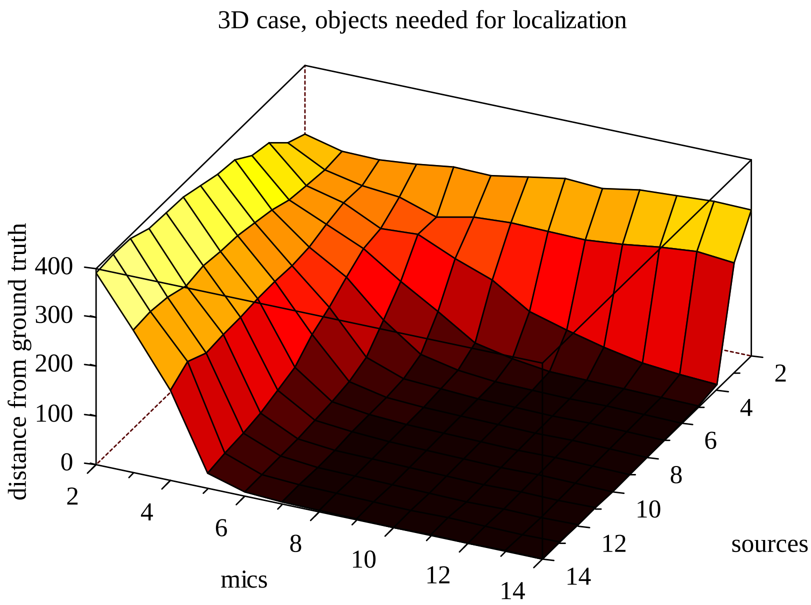

Figure 6: Series of 100 random runs in three dimensions for any object count combination, filtered for successful runs. Given at least 4 sound sources, resp. 5 microphones, RMS errors decrease.

number of sound sources is fixed to four, deploying an increasing number of microphones lets the localization error decrease to zero (Fig. 6).

The two-dimensional case showed similar results, with convergence for four microphones, resp. three sound sources (Fig. 7). These observations correspond to our considerations from Section 1.4 where we predicted the reconstructability of all unknown positions for such numbers of sound sources and microphones.

3 Ellipsoid TDOA method for distant sources

Now we consider the case where the signal origins are very far from the receivers. Under this assumption we develop an approximative approach to reveal distances and angles between a fixed number of receivers, i.e. three receivers for the plane and four receivers for the three-dimensional space. In this special case a smaller number of sound signals is sufficient to compute the relative locations than in the general case. Furthermore, the solution of the problem can be expressed in a closed form.

Once the receiver triangle ABC, resp. tetrahedron $ABCD$ for three dimensions, has been reconstructed we determine the direction of the signal origins. We will now present the two-dimensional case.

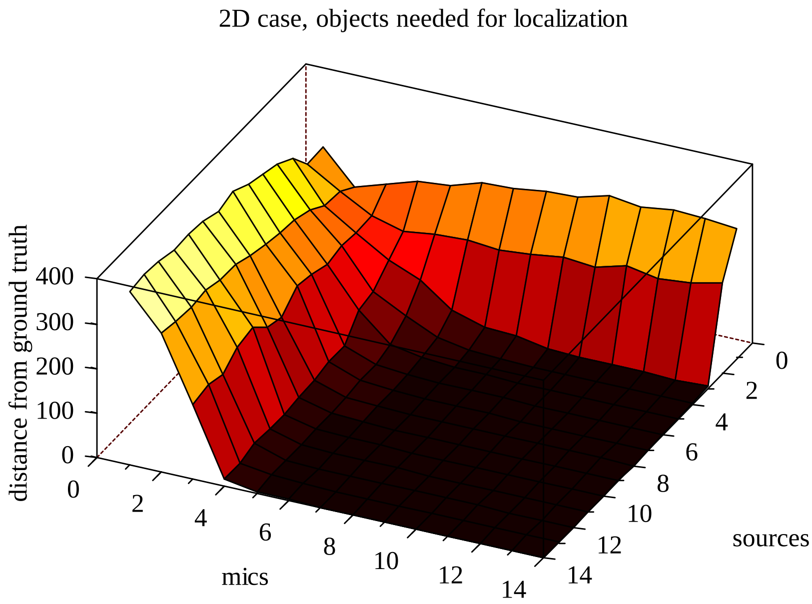

Figure 7: Series of 100 random runs in two dimensions for any object count combination, filtered for successful runs. Given at least 3 sound sources, resp. 4 microphones, RMS errors decrease.

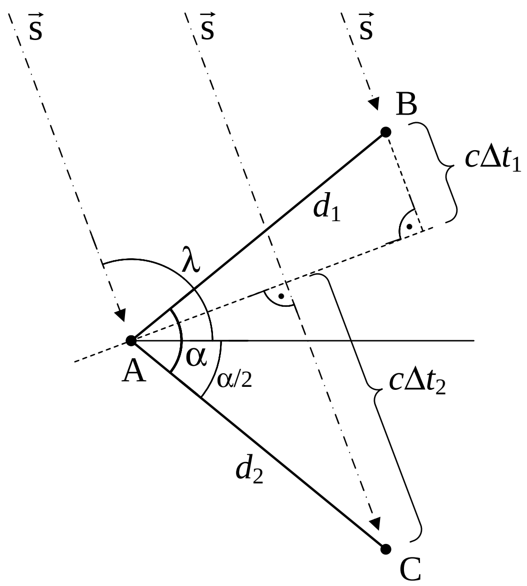

Figure 8: Three receivers A, B, C and a signal on the horizon with direction $\vec{s}$.

3.1 TDOA ellipse

For three receivers A, B, C in the plane and a distant source S the discrete signal is received by the receivers at time points t_A, t_B and t_C, see Fig. 8. Define

$$\begin{aligned} \Delta t_1 &= t_B - t_A \quad (12) \\ \Delta t_2 &= t_C - t_A \quad (13) \end{aligned}$$

where Δt_1 and Δt_2 are the time differences of arrival (TDOA) between A and B, resp. A and C. For $\alpha = \angle_{CAB}$ and using the assumption of infinite distant signal origins we state:

$$\begin{aligned} x := \Delta t_1 &= d_1 \cos\left(\gamma - \frac{\alpha}{2}\right) \quad (14) \\ y := \Delta t_2 &= d_2 \cos\left(\gamma + \frac{\alpha}{2}\right) \quad (15) \end{aligned}$$

where γ denotes the direction of $\vec{s}$ with respect to the bisection of α. Combining the equations we derive the following ellipse equation:

$$x^2 \frac{1}{d_1^2} + y^2 \frac{1}{d_2^2} + xy \frac{-2\cos\alpha}{d_1 d_2} = \underbrace{\frac{1}{2} - \frac{1}{2}\cos 2\alpha}_{\sin^2\alpha} \quad (16)$$

Normalization by division by $\sin^2\alpha$ (under the assumption $\alpha \notin \{0, \pi\}$, i.e. A, B, C are collinear) leads to the ellipse parameters

$$\begin{aligned} a &= \frac{1}{d_1^2 \sin^2\alpha} \quad (17) \\ b &= \frac{1}{d_2^2 \sin^2\alpha} \quad (18) \\ c &= \frac{-2\cos\alpha}{d_1 d_2 \sin^2\alpha} \quad (19) \end{aligned}$$

for $ax^2 + by^2 + cxy = 1$.

For the localization we measure at least three pairs of time differences $(\Delta t_1, \Delta t_2)$ of different distant signal origins. From these points we compute the ellipse equation with parameters a, b, c, see Fig. 9. Then, we use the above equations to compute d_1, d_2, α which can be done by the following equations:

$$\begin{aligned} d_1 &= 2\sqrt{\frac{b}{4ab - c^2}} \quad (20) \\ d_2 &= 2\sqrt{\frac{a}{4ab - c^2}} \quad (21) \\ \alpha &= \arccos \frac{-c}{2\sqrt{ab}} \quad (22) \end{aligned}$$

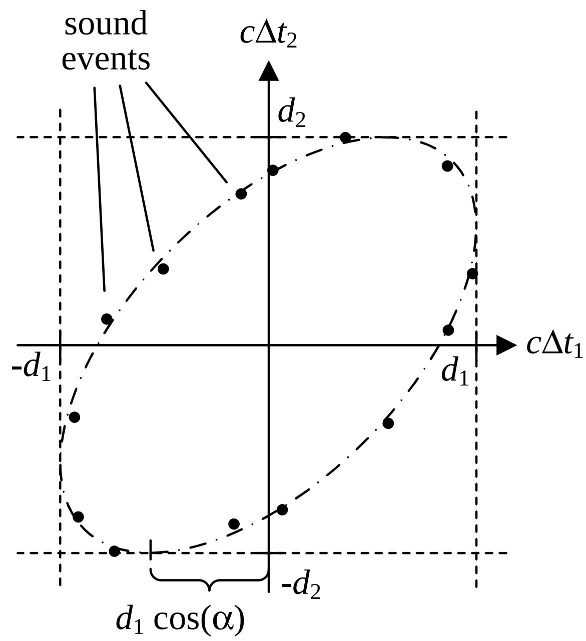

Figure 9: Multiple distant signal sources with time difference pairs $(\Delta t_1, \Delta t_2)$ in two dimensions form an ellipse.

3.2 Linear regression

Three ambient signals are sufficient to find the ellipse for two dimensions. Since ambient radio or sound signals are no scarce resource the additional signals can be used to overcome the inaccuracies caused by imprecise time measurements and other error sources. Given a sufficient number of m signal sources which form a set of (x, y)-tuples we obtain a system of linear equations which is:

$$ax_i^2 + by_i^2 + cx_iy_i = 1 \tag{23}$$

where $1 \leq i \leq m$. We use linear regression to reconstruct the parameters of this ellipse. In matrix notation this is:

$$\underbrace{\begin{pmatrix} x_1^2 & y_1^2 & x_1y_1 \\ \vdots & \vdots & \vdots \\ x_m^2 & y_m^2 & x_my_m \end{pmatrix}}_{\mathbf{Q}} \underbrace{\begin{pmatrix} a \\ b \\ c \end{pmatrix}}_{\mathbf{x}} = \vec{1} \tag{24}$$

If $m > 3$ we use the least squares method and solve:

$$\left(\mathbf{Q}^T\mathbf{Q}\right)\mathbf{x} = \mathbf{Q}^T\vec{1} \tag{25}$$

The ellipse parameters $\mathbf{x}$ are derived as follows:

$$\mathbf{x} = \begin{cases} \mathbf{Q}^{-1} \cdot \vec{1}, & \text{if } m = 3 \\ \left(\mathbf{Q}^T\mathbf{Q}\right)^{-1} \cdot \left(\mathbf{Q}^T\vec{1}\right), & \text{else} \end{cases} \tag{26}$$

Then, we use the equations of the previous subsection to compute the geometry of the triangle ABC. Since the assumption of infinitely far senders is not realistic this approach results in an approximative solution of the problem.

However, this is the best one can offer if only three signal sources are available, since the problem for three general signal positions is under-defined as we pointed out in the last section. Later on, we present simulations which indicate that the approximation behaves well if the signals are a small constant factor farer than the longest edge of the receiver triangle.

3.3 Three dimensional case

The three dimensional case is similar. For a source S and receivers A, B, C let x, y, z be the time differences of B, C and D with respect to A. The source angles are $\gamma_B = \angle_{BAS}$, $\gamma_C = \angle_{CAS}$, $\gamma_D = \angle_{DAS}$ which lead to the equations:

$$\begin{aligned} x &= d_B \cos\gamma_B \\ y &= d_C \cos\gamma_C \\ z &= d_D \cos\gamma_D \end{aligned}$$

For angles ϕ_C, ϕ_D and λ_D defined as shown in Fig. 10 one can derive the following ellipsoid equation:

$$\left(\frac{x}{d_B}\right)^2 + \left(-\frac{\cos\phi_C}{d_B \sin\phi_C}x + \frac{y}{d_C \sin\phi_C}\right)^2 + \left(-\frac{\cos\phi_D \sin\phi_C + \cos\phi_C \sin\phi_D \cos\lambda_D}{d_B \sin\phi_C \sin\phi_D \sin\lambda_D}x + \frac{\cos\lambda_D}{d_C \sin\phi_C \sin\lambda_D}y + \frac{z}{d_D \sin\phi_D \sin\lambda_D}\right)^2 = 1$$

$$(ax)^2 + (bx + cy)^2 + (dx + ey + fz)^2 = 1$$

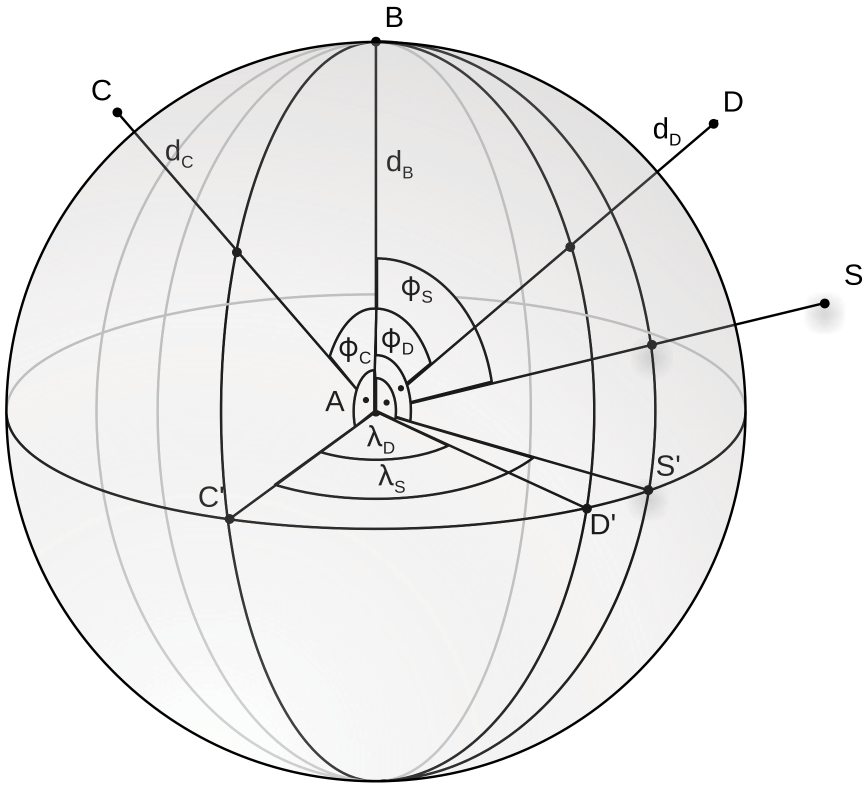

Figure 10: Given 4 microphones A, B, C and D and a distant sound source S in three dimensions. Distances are d_A, d_B, d_C and d_D, angles are ϕ_C, ϕ_D and λ_D, which uniquely determine a tetrahedron.

where

$$
\begin{aligned}
a &= \frac{1}{d_B} \\
b &= -\frac{\cos\phi_C}{d_B \sin\phi_C} \\
c &= \frac{1}{d_C \sin\phi_C} \\
d &= -\frac{\cos\phi_D}{d_B \sin\phi_D \sin\lambda_D} - \frac{\cos\phi_C \cos\lambda_D}{d_B \sin\phi_C \sin\lambda_D} \\
e &= \frac{\cos\lambda_D}{d_C \sin\phi_C \sin\lambda_D} \\
f &= \frac{1}{d_D \sin\phi_D \sin\lambda_D}
\end{aligned}
$$

Given a, b, c, d, e, f one can compute the angles and distances by:

$$
\begin{aligned}
d_B &= \frac{1}{a} \\
\tan\lambda_D &= \frac{c}{e} \\
\tan\phi_C &= -\frac{a}{b} \\
d_C &= \frac{1}{c \sin\phi_C} \\
\frac{1}{\tan\phi_D} &= -d\ d_B \sin\lambda_D - \frac{\cos\lambda_D}{\tan\phi_C} \\
d_D &= \frac{1}{f \sin\phi_D \sin\lambda_D}
\end{aligned}
$$

The ellipsoid parameters can be computed by regression analogous as in the two-dimensional case. For three dimensions a minimum of $m = 6$ signal sources is needed to be uniquely solved. If $m > 6$ the least squares method is used as described above.

Remark: This solution leads to a duplicate answer for ϕ_D. It can be avoided by calculating ϕ_C and afterwards swap the y- and z-axis. Now recalculate the parameters a – f and use the formula for ϕ_C again, which now equals (unique) ϕ_D.

3.4 Simulation

We have tested the accuracy of this approximation algorithm with a computer algebra system. To simulate measurement errors a probabilistic Gaussian error model has been added to each time stamp.

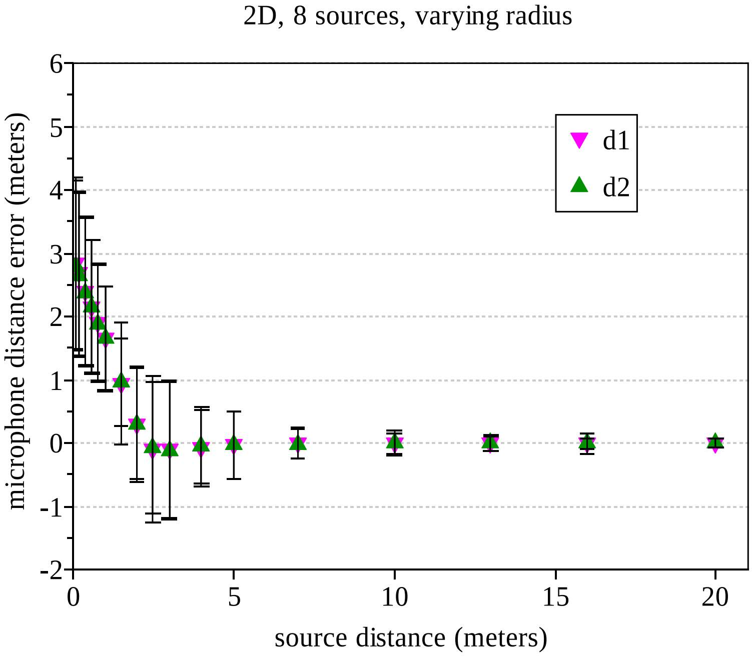

Figure 11: Increasing sound source distance above 4 m results in distance approximation errors below 0.1 m for two dimensions.

A simulation run consists of a number of sound sources arbitrarily arranged on a circle for two dimensions and a sphere for three dimensions with a fixed radius around A. The three microphones in the two-dimensional case are positioned on a circle with fixed radius of 2.3 m forming a triangle of about 4 m of edge length. In the three-dimensional case the four microphones are placed on a sphere hull of the same radius forming a tetrahedron.

For each radius of 0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 7.0, 10, 13, 16 and 20 meters a series of 1,000 tests with 8 sound sources has been run. The distance results d_1 and d_2 and the angle α between A and B are subtracted from the real values, which are read from the triangle properties. For the three-dimensional case these are distances d_B, d_C and d_D and angles ϕ_C, ϕ_D and λ_D. Failing runs are produced if the approximated quadratic equation does not describe an ellipse or ellipsoid. For successful runs we calculated the average and the standard deviation of the distance and angle differences.

The results show a systematic under-estimation of the distances between microphones for short ranges which improves after the perimeter of the microphones has been left at about 5 m. The angle errors show high variance within the perimeter of the microphones which stabilizes quickly upon leaving it, at a range of 5 m. Failing localizations occur especially if the sound source radius equals the microphone radius with up to 4%, but the rate drops quickly to below 1%, see Fig. 11 – 13 for two dimensions and Fig. 14 – 16 for three dimensions.

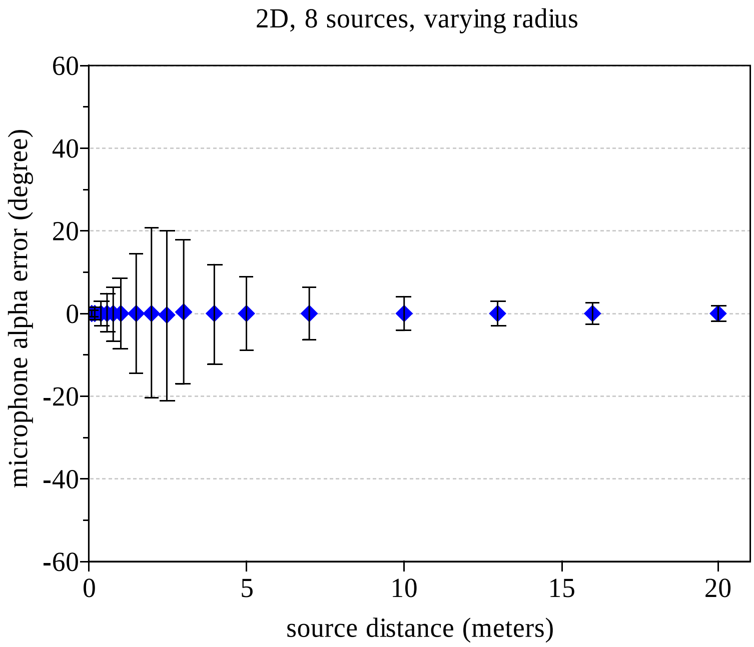

Figure 12: Increasing sound source distance above 4 m results in angle approximation errors below 2° for two dimensions.

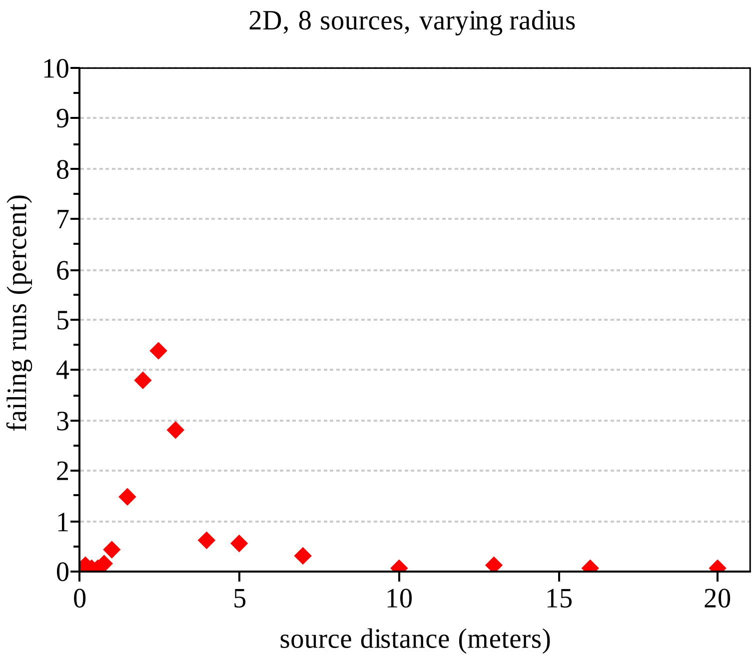

Figure 13: After reaching a maximum at a sound source distance of 2.5 m the failure rate drops below 1 % for greater distances.

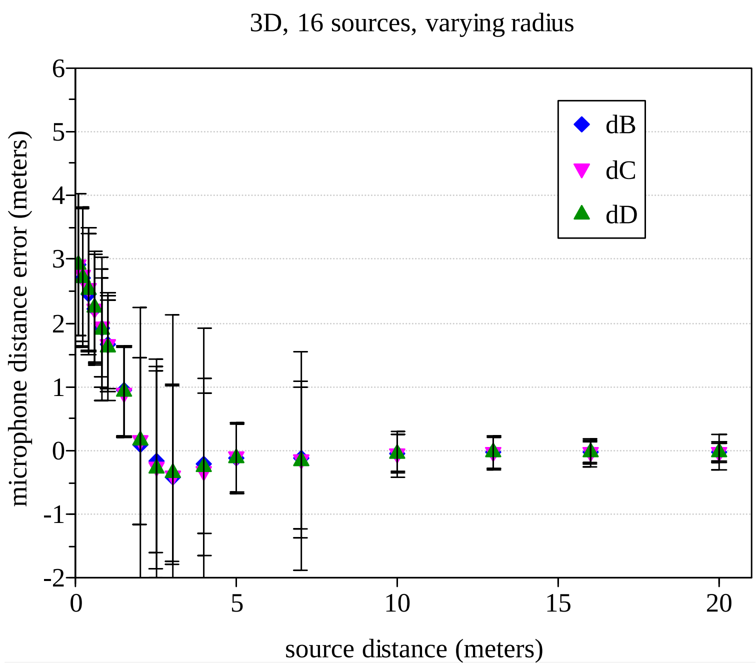

Figure 14: Increasing sound source distance above 4 m results in distance approximation errors below 0.2 m for three dimensions.

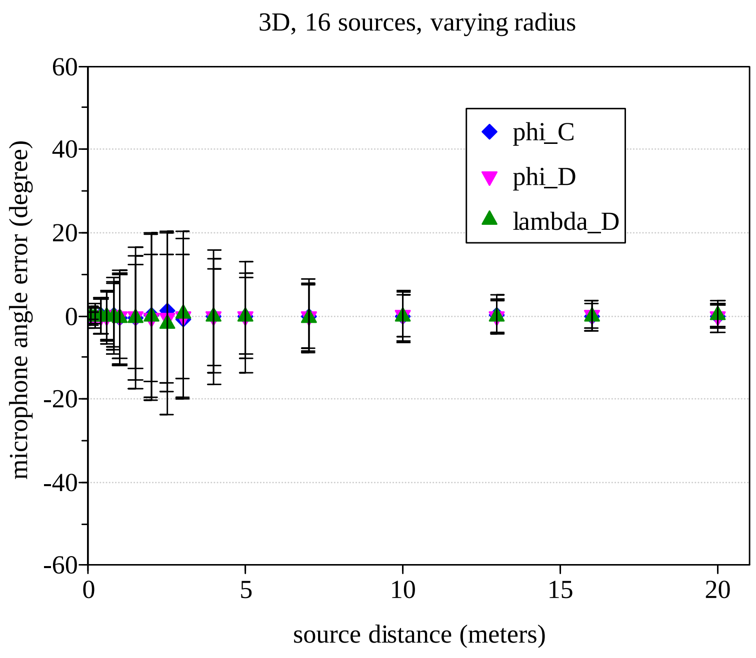

Figure 15: Increasing sound source distance above 4 m results in angle approximation errors below 2° for three dimensions.

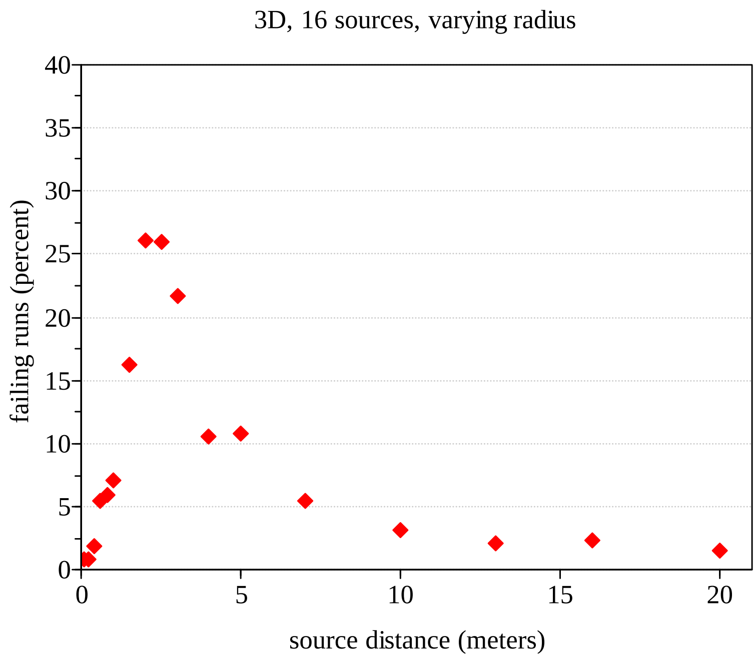

Figure 16: After reaching a maximum of 25 % at a sound source distance of 2.5 m the failure rate drops below 5 % for greater distances.

A stress test was run to observe the behavior of the approximation in case of runtime variances (Fig. 17 – 19). Distant sound sources were assumed (radius of 1,000 m) and the gaussian runtime error was increased to a standard deviation of 0.2, 0.4, 0.7, 1.0, 1.3, 1.6 and 2.0 ms.

Results show a slight over-estimation of the microphone distances and a moderate increase in angular variance. Failures increase to about 5 %. However, in that narrow setup of 4 m edge length an error of 2 ms equals about 70 cm spatial error. The time difference measures in this magnitude drawn as x/y-plot are hardly recognizable as an ellipse any more (Fig. 20). In our real world experiments we observed runtime errors of about 0.2 ms of standard deviation, which is way below the errors we induced here.

3.5 Summary

Our considerations about the degrees of freedom pointed out that position reconstruction without any given anchors cannot be done with less than four receivers in two dimensions and with less than five receivers in three dimensions. However, the approximation scheme enables us to state some propositions about the receiver positions and the direction of the signal sources even with three, resp. four receivers.

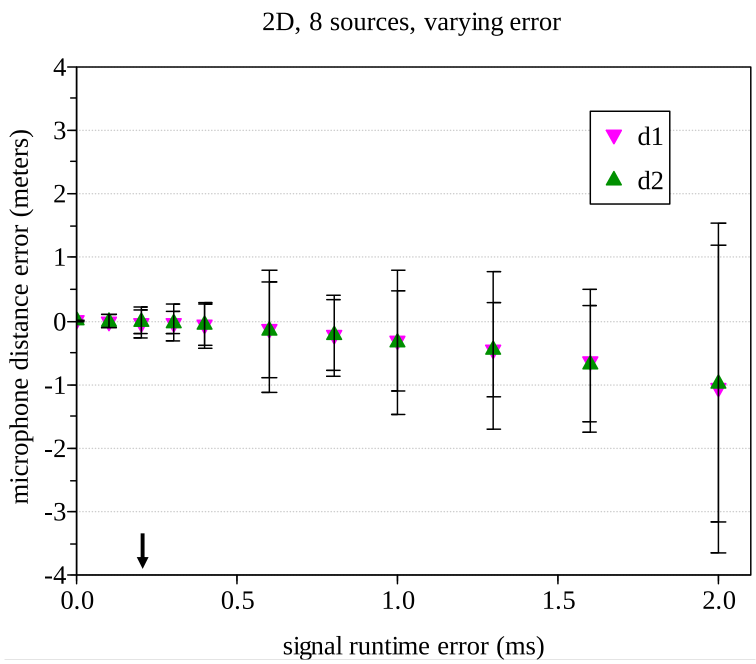

Figure 17: Increasing TDOA error leads to higher variance in distance approximation and a systematic over-estimation of distances for two dimensions. In reality values of about 0.2 ms were observed.

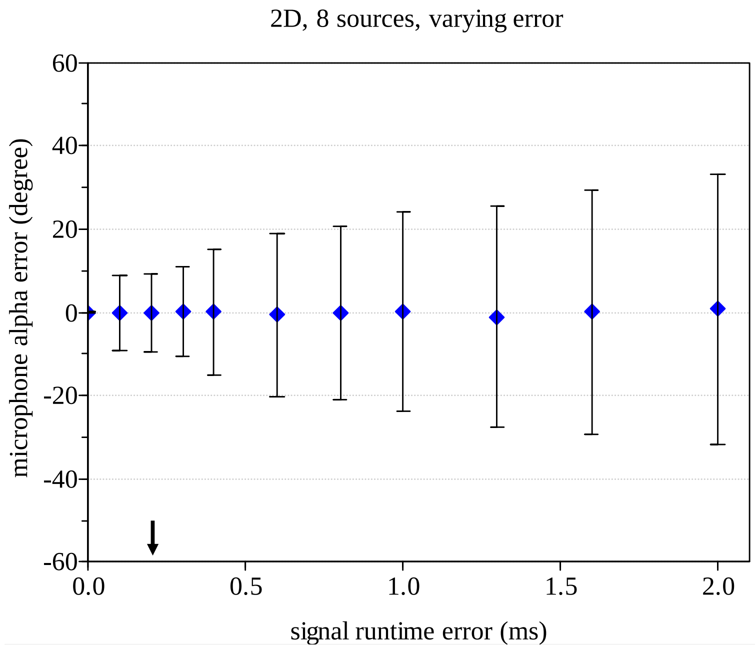

Figure 18: Increasing TDOA error leads to higher variance in angle approximation for two dimensions.

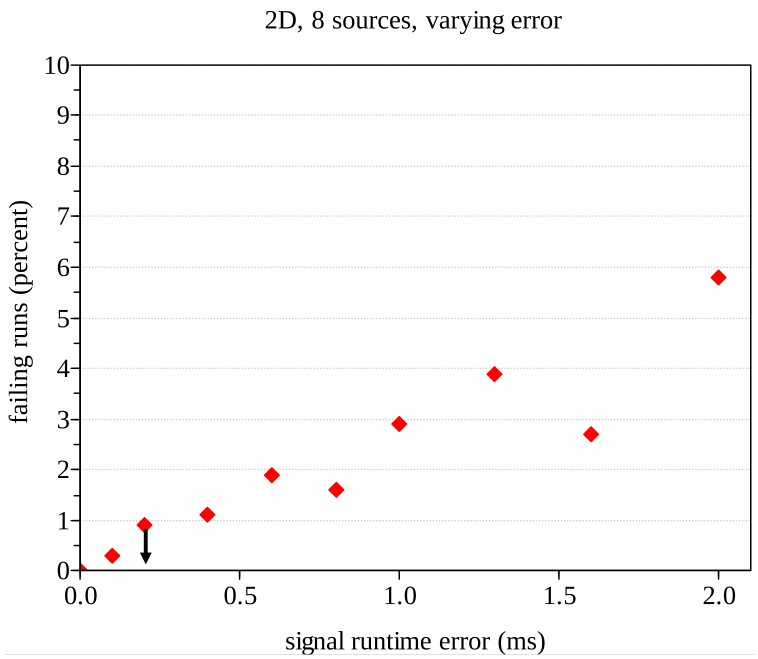

Figure 19: Increasing TDOA error leads to an increased failure rate for two dimensions.

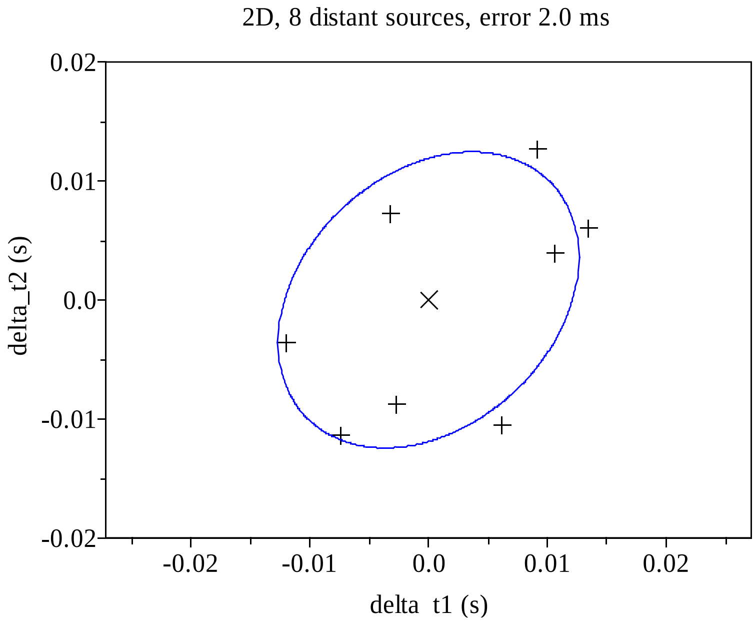

Figure 20: Eight signals with Gaussian error of $\sigma = 2\,\text{ms}$ received in two-dimensional space. The elliptical form is hardly recognizable due to this high error.

The technique directly benefits from an increased number of signal events. These are cheaply available in many environments. Then our technique becomes very robust, even for noisy data.

Simulation and real world tests suggest that our assumption of infinitely remote signal sources is not far-fetched. The parallax decreases quickly, as soon as we are outside the receiver perimeter. It allows to use the approximation even for close-ranged scenarios.

The approximation scheme fails if receiver positions collapse on a line or a plane. In this case, the time differences form a line of which no ellipse can be extracted. However, this singular case can be detected and treated particularly.

In some cases of noisy data we found a slight, systematic over-estimation of receiver distances and of angles. For the stress test runs this resulted in higher variance. Visual analysis of the time differences showed that the resulting ellipse does not fit the corpus of the noisy data. This seems to be a result of deficient ellipse regression.

4 Real world experiments

We have tested this theoretical approach in several real world experiments. For this we use a network with laptops as network nodes. Our software establishes a TCP/IP-communication via local area network (LAN) between several laptops and assures precise time synchonisation. With the built-in laptop microphones we record sound signals. The audio track is filtered for sharp sound events, like clapping or finger snapping and their points in time are determined. As a peculiar mark for a sound event we use the moment when the signal rises above a environment noise dependant threshold for the first time.

Threshold comparisons showed to be the robustest approach with only little drawbacks in precision. Maximum searches, either directly or derivative (edge detection) showed to be slightly more precise but prove to be ambiguous with devastating results in cases when hosts chose different maxima.

In a client server model these signals are collected at the server which computes the relative locations using the algorithms described before. Based on priority IDs this server is elected in the beginning and is re-elected every time a node enters or leaves the network.

4.1 Time synchronisation

Common TDOA localization features precise synchronisation among receivers. While unsynchronized localization is generally possible, time synchronization reduces the number of required sound events. Using the assigned master host as a time reference, all clients synchronize to its clock. This is done by a series of pinging the master which answers with its current time. This timestamp is corrected by 1/2 RTT (round trip time), assuming the network packet took the same runtime in both directions. The obtained timestamps are filtered for high RTT (outliers), which result from network jitter.

Our experiments showed that clock drift correction is essential even with the utilized high precision event timer (HPET). Although running with accurately constant speed, drift rates between HPET clocks of 0.03 % were observed, which is very high. We obtain both time offset and clock drift between client and master by linear regression of the timestamp set.

4.2 Experiments

Our real world tests took place in a large lecture hall. We used a scenario of 6 laptops positioned in an ellipse like formation of dimensions of $17\,\mathrm{m} \times 13\,\mathrm{m}$. They all were connected over an ethernet based LAN with one switch, so the timestamping software running on each laptop could communicate with each other and provide precise time synchronization among all devices. With their built-in microphones they were to record any incoming sound event and assign precise timestamps to prominent signals using audio processing.

A little girl kindly agreed to act as a noisemaker utilizing a toy gun and a metal spoon on a glass bottle. She was free to choose positions making noise, but was advised not to make the noise signals from identical locations.

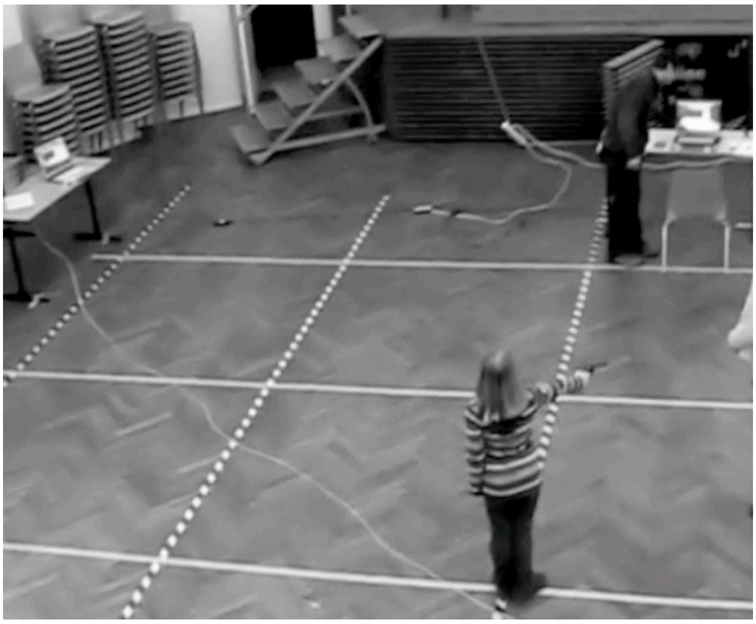

Figure 21: Girl with toy gun and two of the five notebook computers to be located from 12 shots at unknown positions.

The positions of the microphones of the laptops were charted up to a precision of 2 cm. The positions of the sound signals were marked with plastic caps on the floor, where we had drawn a grid to facilitate position mapping. Since the girl was free to move we could chart the girl's sound signals only to a precision of 10 cm. We performed several tests with variations of noise types and sound event numbers. The sound events recorded during the tests were assigned a timestamp. The Cone Alignment algorithm got these timestamps as the only input and computed the relative locations of microphones and sound signals.

4.2.1 Position reconstruction

We start with the toy gun scenario, see Fig. 21. Here only five computers were used. We obtained 12 sound signals at 11 positions where only 11 sound signals have been toy gun shots. One was accidentally produced by the placement of a plastic cap.

All real position marks were plotted with the obtained positions, see Fig. 22. The mean distance error of all microphones and sound sources after SVD alignment is 0.270 m with a standard deviation of 0.165 m – less than a laptop width.

In a subsequent test, we were checking for systematic errors by producing many signals at one position. Series of 10 signals of a metal spoon on a glass bottle were made at seven positions and marked on the floor. Again, we removed timestamps which were not recorded

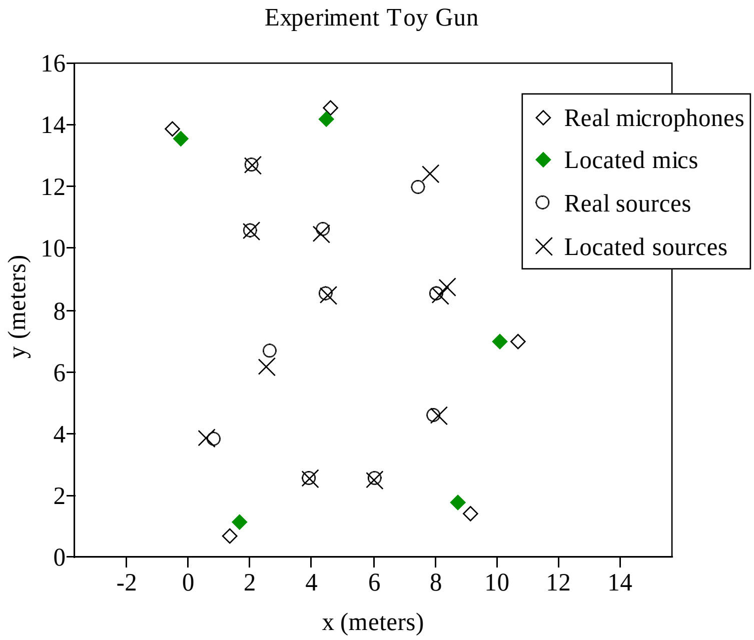

Figure 22: Five notebook computers located by receiving 12 sound signals produced by a girl with a toy gun.

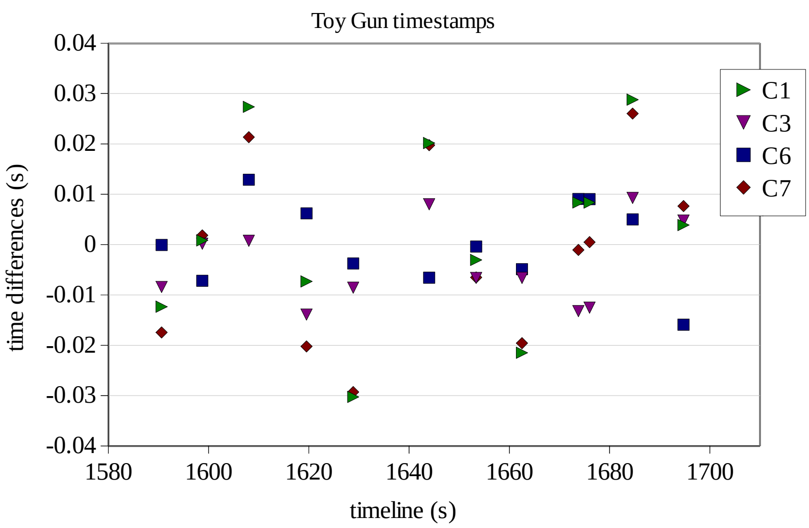

Figure 23: Timestamps of the toy gun experiment relative to network node C5.

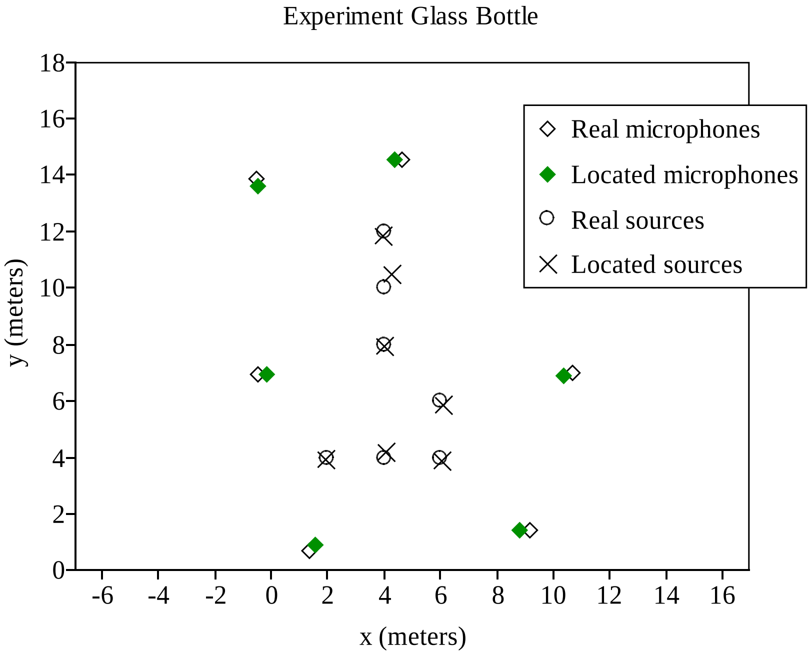

Figure 24: Scenario with 6 laptops. Seven series of pings on a glass bottle.

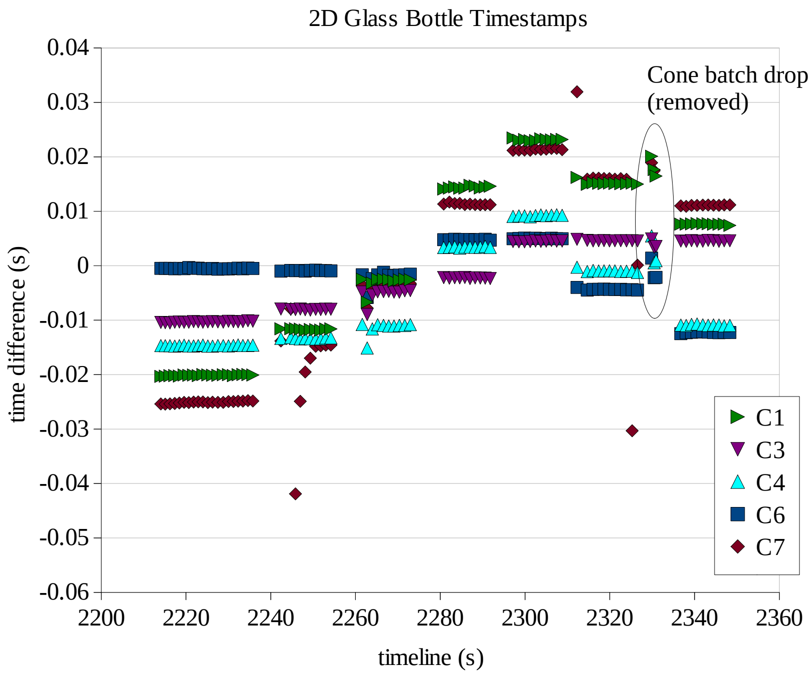

Figure 25: Glass bottle experiment, timestamps of 5 laptops relative to laptop C5. C7 shows a few obvious outliers.

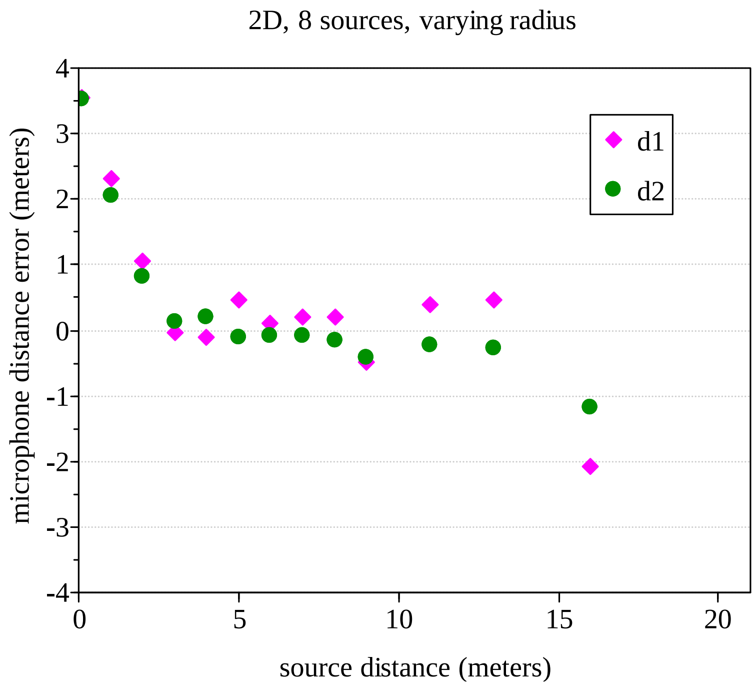

Figure 26: Distance errors of d_1 and d_2 for the Ellipsoid method experiment.

at all receivers. Timestamps for one position were averaged and fed into the localization software. We received a mean distance between microphones and sound sources of 0.223 m and a standard deviation of 0.097 m.

On analyzing the recorded timestamp set, we found some extreme outliers. After manually removing them this improved the mean error to 0.098 m with a standard deviation of 0.052 m – which is the precision we used to position the marker caps. We are planning to automatize this filtering for future versions.

In our real world experiments we did not encounter the local minima issue as we had strictly overdetermined scenarios.

4.2.2 Ellipsoid TDOA method

One experiment aimed at testing the Ellipsoid TDOA method. We positioned three laptops A, B and C in a corner of the lecture room, residing on a circle with radius 2.3 m, as in the simulations. The distances between them were $d_{AB} = 4.30$ m, $d_{AC} = 4.14$ m and $d_{BC} = 3.47$ m, which results in $\angle_{CAB} = 48.6°$. Now we covered the lecture hall generating sound signals with fixed distances to the circle center, which were 0.1, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 11, 13 and 16 m.

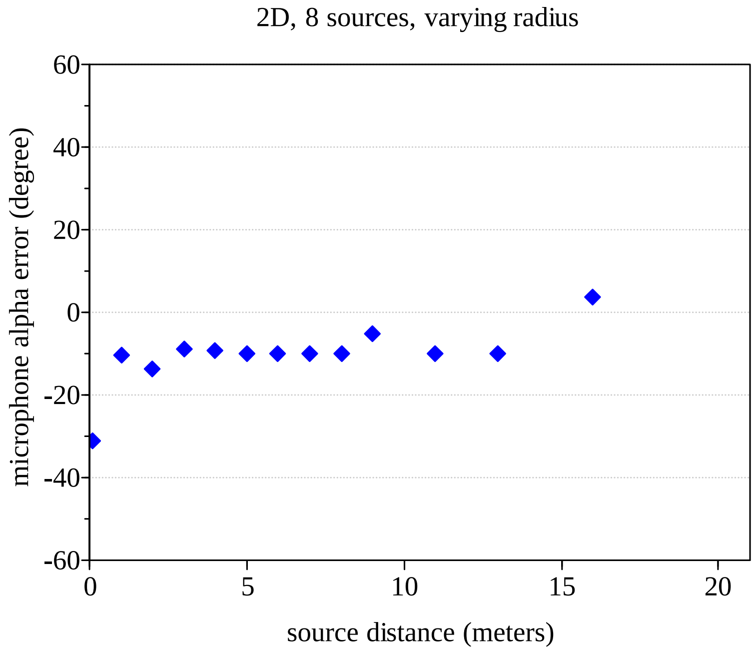

Figure 27: Angle error of α for the Ellipsoid method experiment.

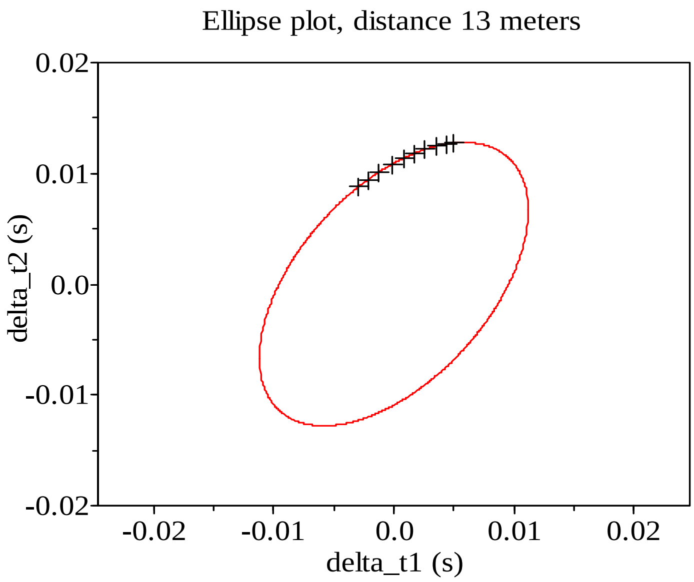

Figure 28: Time differences as a x/y-plot for the approximation experiment for two dimensions. Sound signals from a distance of 13 m arrive from only one direction.

The obtained data was grouped by center distances and timestamps were filtered for outliers. Those with differences of more than 20 ms, which cannot occur in that small triangle, were removed.

Evaluation showed a good convergence of the microphone distance approximations d_1 and d_2 at a circle radius of 4 m and above (Fig. 26). Errors fall below 0.5 m. Angle α resides at about 58°, which is an over-estimation of 10° (Fig. 28).

With increasing sound signal distance results degrade, which we attribute to the narrowing sector of sound origins that come from only one direction, thus making it harder to describe an ellipse, see Fig. 28. This seems to be a drawback of the technique. Obviously it needs signals from different angles to reconstruct the ellipse.

5 Conclusions

To our knowledge we have considered for the first time the problem of relative localization of nodes in a computer network solely based on ambient signals. There is absolutely no knowledge available about the received signals except that they are received by all nodes of the network and they can be distinguished from each other. For the network no anchor nodes are given. It uses the fact that the network nodes are synchronized and that the ambient signals originate from punctual origins. Furthermore, we assume that they travel with a constant speed on a direct line. To our knowledge this problem has never been addressed before.

For this novel problem we present two solutions. The first, Iterative Cone Alignment, solves the general case if the number of signals or the number of receivers is large enough. It is based on a physical spring-mass particle simulation to find an approximate solution of a series of equations of polynomial order four. Simulations show that there is a chance of running into local minima but this probability decreases when more ambient signals are collected.

The second, the Ellipsoid TDOA method, presents a closed form solution if the signals are far away. Then, a smaller number of receivers and signals suffices to find appropriate positions. From the simulations we know that the approximation error behaves very benign.

For a real-world experiment we have implemented our algorithms and designed a networking protocol. Then, a small girl produced some sound signals from positions of her choice using a toy gun or a glass bottle. With the built-in microphones the notebooks in this room were able to compute their relative distances within an error less than the size of the notebook computers.

All of our findings are based on the existence of a communication network of at least five nodes which can be reduced to four nodes when the ambient signals are from a distant, or three nodes if the nodes and the far-distant signals are in the plane.

It is a well-known technique that antenna arrays can compute the direction of signals using the phase differences at the antennas. However, in our settings we use an antenna array where the relative positions are not known in the beginning. One might claim that we have generalized the localization problem known from antenna arrays.

Since we addressed the approach of the relative localization from ambient signals for the first time there is a lot of potential for improvements and applications. The results presented here show that using the ubiquitous resources of ambient signals gives us an easy and accurate way to locate network nodes.

5.1 Applications

We will shortly point out a small number of applications of this novel approach.

First of all the sound localization method is the method of choice to compute the location of computers in a room. The computers do not need to emit sound signals and the noises produced by users are sufficient to compute locations. Of course, also the position of users can be detected. Since most modern PCs have built-in microphones and are connected to a LAN this method can be applied.

Another straight-forward application is its use in sensor networks. Microphones are very cheap sensors and localization of sensor nodes is a time consuming task. Using our scheme it is sufficient to produce sharp sound signals at the beginning or the end of a field test to determine the locations of the sensors. The combination of the results can be done on-line by transferring the timestamps or off-line after recollection of the sensor nodes.

Future applications involve the use of radio signals instead of sound signals. As a precondition special hardware is needed to determine the exact time points of radio signals because of the much larger speed of light. Given such hardware it is possible to compute the relative positions of network nodes like notebook computers, mobile phones, tablet computers or PDAs by using ambient radio signals coming from WLAN base stations, radio or TV broadcast, TV satellites or lightnings. Of course such a localization method must be combined with anchors which give absolute locations. Such anchors could be the positions of some senders (e.g. encrypted WLAN stations) or receivers (e.g. special infrastructure devised for such a system).

We want to point out that our method does not use the information when or where a signal is produced. The only information needed is when it is received at a sender and that this signal cannot be mixed up with other signals. Therefore, in our approach the participants of the location system do not need to understand the signals. For example an encrypted GPS signal from a satellite of unknown positions may also be a useful input for our algorithm, likewise a TV satellite or the mobile phone of a passer-by. This clearly separates our approach from the prevalent approach which uses the information of time of flight, i.e. time of arrival (TOA).

5.2 Future work

It is very obvious that both approximation methods can be combined. In future work we will use the Ellipsoid TDOA method to construct a triangulation of the network which might decrease the chance that the Iterative Cone Alignment runs into a local minimum. Besides, other methods to recover from such local minima will be considered. In another combination the Ellipsoid TDOA method might serve as a filter for the timestamps used in the Cone Alignment since the Ellipsoid scheme is more robust against outliers.

We have also seen some room for improvement in the approximation of the TDOA ellipse. While our regression minimizes the error "in some least squares sense" [24], there are more sophisticated techniques available like *geometric fit* proposed by Gander et al. [24].

Further research will involve the use of non-discrete continuous signals, e.g. voices, traffic noise or analogous radio signals. By testing for best overlaps of such signals it should be possible to compute a time difference analogously to sharp signals. This would dramatically increase the information basis of the algorithms. Another interesting research topic is to consider the mobility of continuous signal sources and receivers.

In the future we will implement this sound based approach on sensor networks and mobile phones. Furthermore, we will design and test special hardware devices which will facilitate relative localization from ambient signals.

References

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

[2] Ralph D. Hippenstiel, Timothy Haney, and Tri T. Ha. Improvement of the Time Difference of Arrival (TDOA) estimation of GSM Signals Using Wavelets. *NPS-EC-00-08, Naval Postgraduate School*, 2000.

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

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

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

[7] 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.

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

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

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

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

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

[13] 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.

[14] 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.

[15] 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.

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

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

[18] 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.

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

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

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

[22] 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.

[23] 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.

[24] W. Gander, G.H. Golub, and R. Strebel. Least-Square Fitting of Circles and Ellipses. *BIT Numerical Mathematics*, 34(4):558–578, Dec. 1994.

6 Appendix

6.1 Reformulation as Optimization Problem with Polynomials of Degree 4

Given $t_{i,j} = t_{M_i,S_j}$ we have the constraints

$$c^2(t_i - t_{i,j})^2 = (x_i - x'_j)^2 + (y_i - y'_j)^2 + (z_i - z'_j)^2$$

for unknown variables $t_i, x_i, y_i, z_i, x'_j, y'_j, z'_j$ for $i \in \{1, \dots n\}$ and $j \in \{1, \dots, m\}$. We consider the case where the system is over-restraint. Then, finding a good solution is to minimize the polynomial of degree 4

$$\sum_{i,j} \left(c^2(t_i - t_{i,j})^2 - (x_i - x'_j)^2 + (y_i - y'_j)^2 + (z_i - z'_j)^2 \right)^2 .$$

Since even the quadratic optimization problem where the function $f(x) = \frac{1}{2}x^T Q x + c^T$ should be minimized is known to be NP-hard for general Q there is even less hope for solving optimization problems of degree 4.

6.2 Newton approximation

A standard technique of solving a system of differentiable nonlinear equations is Newton's method. A well-known approach retrieves one signal source with four given receivers. For every receiver $i = 1..4$ the equation is stated:

$$f_i(x,\, y,\, z,\, t) = c(t - t_i) - \sqrt{(x - x_i)^2 + \ldots + (z - z_i)^2} \tag{27}$$

At a starting position $(x_0,\, y_0,\, z_0,\, t_0)$ a Taylor approximation is used and solved for:

$$\vec{b} \;=\; A\vec{x}$$

$$\begin{bmatrix} -f_1(x_0,\, y_0,\, z_0,\, t_0) \\ -f_2(x_0,\, y_0,\, z_0,\, t_0) \\ -f_3(x_0,\, y_0,\, z_0,\, t_0) \\ -f_4(x_0,\, y_0,\, z_0,\, t_0) \end{bmatrix} = \begin{bmatrix} \frac{\mathrm{d}f_1}{\mathrm{d}x} & \frac{\mathrm{d}f_1}{\mathrm{d}y} & \frac{\mathrm{d}f_1}{\mathrm{d}z} & \frac{\mathrm{d}f_1}{\mathrm{d}t} \\ \frac{\mathrm{d}f_2}{\mathrm{d}x} & \frac{\mathrm{d}f_2}{\mathrm{d}y} & \frac{\mathrm{d}f_2}{\mathrm{d}z} & \frac{\mathrm{d}f_2}{\mathrm{d}t} \\ \frac{\mathrm{d}f_3}{\mathrm{d}x} & \frac{\mathrm{d}f_3}{\mathrm{d}y} & \frac{\mathrm{d}f_3}{\mathrm{d}z} & \frac{\mathrm{d}f_3}{\mathrm{d}t} \\ \frac{\mathrm{d}f_4}{\mathrm{d}x} & \frac{\mathrm{d}f_4}{\mathrm{d}y} & \frac{\mathrm{d}f_4}{\mathrm{d}z} & \frac{\mathrm{d}f_4}{\mathrm{d}t} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}$$

for the update vector $\vec{x}$ by inverting the Jacobian matrix A that contains the derivations for 4 equations and dimensions x, y, z and t:

$$\begin{aligned}
\frac{\mathrm{d}f_n}{\mathrm{d}x} &= \frac{x - x_n}{\sqrt{(x-x_n)^2 + (y-y_n)^2 + (z-z_n)^2}} \\
\frac{\mathrm{d}f_n}{\mathrm{d}y} &= \frac{y - y_n}{\sqrt{(x-x_n)^2 + (y-y_n)^2 + (z-z_n)^2}} \\
\frac{\mathrm{d}f_n}{\mathrm{d}z} &= \frac{z - z_n}{\sqrt{(x-x_n)^2 + (y-y_n)^2 + (z-z_n)^2}} \\
\frac{\mathrm{d}f_n}{\mathrm{d}t} &= c
\end{aligned}$$

$$\vec{x} = A^{-1}\vec{b} \tag{28}$$

We extend this approach for n receivers and m signal sources and derive for mn equations and $3n + 4m$ dimensions:

$$A = \begin{bmatrix}
\frac{\mathrm{d}f_{11}}{\mathrm{d}x_{\mathbf{S}_1}} & \cdots & \frac{\mathrm{d}f_{11}}{\mathrm{d}t_{\mathbf{S}_m}} & \frac{\mathrm{d}f_{11}}{\mathrm{d}x_{\mathbf{M}_1}} & \cdots & \frac{\mathrm{d}f_{11}}{\mathrm{d}z_{\mathbf{M}_n}} \\
\frac{\mathrm{d}f_{12}}{\mathrm{d}x_{\mathbf{S}_1}} & \cdots & \frac{\mathrm{d}f_{12}}{\mathrm{d}t_{\mathbf{S}_m}} & \frac{\mathrm{d}f_{12}}{\mathrm{d}x_{\mathbf{M}_1}} & \cdots & \frac{\mathrm{d}f_{12}}{\mathrm{d}z_{\mathbf{M}_n}} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\mathrm{d}f_{nm}}{\mathrm{d}x_{\mathbf{S}_1}} & \cdots & \frac{\mathrm{d}f_{nm}}{\mathrm{d}t_{\mathbf{S}_m}} & \frac{\mathrm{d}f_{nm}}{\mathrm{d}x_{\mathbf{M}_1}} & \cdots & \frac{\mathrm{d}f_{nm}}{\mathrm{d}z_{\mathbf{M}_n}}
\end{bmatrix} \tag{29}$$

Update vector $\vec{x}$ contains all $(3n + 4m)$ unknown variables, function vector $\vec{b}$ consists of mn functions:

$$\begin{aligned}
\vec{x} &= (x_{\mathbf{S}_1}, \ldots, t_{\mathbf{S}_m}, x_{\mathbf{M}_1}, \ldots, z_{\mathbf{M}_n})^T \\
\vec{b} &= \begin{pmatrix}
c(t_{\mathbf{M}_1,\mathbf{S}_1} - t_{\mathbf{S}_1}) & - & |\mathbf{M}_1 - \mathbf{S}_1| \\
\vdots & & \vdots \\
c(t_{\mathbf{M}_1,\mathbf{S}_m} - t_{\mathbf{S}_m}) & - & |\mathbf{M}_1 - \mathbf{S}_m| \\
c(t_{\mathbf{M}_2,\mathbf{S}_1} - t_{\mathbf{S}_1}) & - & |\mathbf{M}_2 - \mathbf{S}_1| \\
\vdots & & \vdots \\
c(t_{\mathbf{M}_n,\mathbf{S}_m} - t_{\mathbf{S}_m}) & - & |\mathbf{M}_n - \mathbf{S}_m|
\end{pmatrix}
\end{aligned}$$

The resulting equation system can be overdetermined for a sufficient number of sources and microphones, so we use the least squares method:

$$A^T\vec{b} = A^T A\vec{x} \tag{30}$$

Instead of computing the pseudo-inverse of $A^T A$ we solve for $\vec{x}$ directly using LU decomposition, which shows better numerical stability.