sensors

MDPI

Article

Acoustic Indoor Localization Augmentation by Self-Calibration and Machine Learning

Joan Bordoy [1,*,†,‡], Dominik Jan Schott [2,†,‡], Jizhou Xie [1,‡], Amir Bannoura [3,‡], Philip Klein [1,‡], Ludwig Striet [1,‡], Fabian Hoeflinger [4,‡], Ivo Haering [4,‡] and Leonhard Reindl [2,‡] and Christian Schindelhauer [1,‡]

[1] Department of Computer Science (IIF), University of Freiburg, 79110 Freiburg, Germany; jizhou.xie@outlook.com (J.X.); philipkln.91@gmail.com (P.K.); ludwig.striet@googlemail.com (L.S.); schindel@informatik.uni-freiburg.de (C.S.)

[2] Department of Microsystems Engineering (IMTEK), University of Freiburg, 79110 Freiburg, Germany; dominik.jan.schott@imtek.uni-freiburg.de (D.J.S.); reindl@imtek.uni-freiburg.de (L.R.)

[3] Department of Software Engineering, Bethlehem University, P.O. Box 11407, Bethlehem, 92248 Jerusalem, Palestine; abannoura@bethlehem.edu

[4] Fraunhofer Institute for Highspeed Dynamics, Ernst-Mach-Institute (EMI), 79104 Freiburg, Germany; Fabian.Hoeflinger@emi.fraunhofer.de (F.H.); ivo.haering@emi.fraunhofer.de (I.H.)

[*] Correspondence: bordoy@informatik.uni-freiburg.de; Tel.: +49-761-203-7070

[†] Current address: Georges-Koehler-Allee 51, 79110 Freiburg im Breisgau, Germany.

[‡] All authors contributed equally to this work.

Received: 17 December 2019; Accepted: 14 February 2020; Published: 20 February 2020

Abstract: An acoustic transmitter can be located by having multiple static microphones. These microphones are synchronized and measure the time differences of arrival (TDoA). Usually, the positions of the microphones are assumed to be known in advance. However, in practice, this means they have to be manually measured, which is a cumbersome job and is prone to errors. In this paper, we present two novel approaches which do not require manual measurement of the receiver positions. The first method uses an inertial measurement unit (IMU), in addition to the acoustic transmitter, to estimate the positions of the receivers. By using an IMU as an additional source of information, the non-convex optimizers are less likely to fall into local minima. Consequently, the success rate is increased and measurements with large errors have less influence on the final estimation. The second method we present in this paper consists of using machine learning to learn the TDoA signatures of certain regions of the localization area. By doing this, the target can be located without knowing where the microphones are and whether the received signals are in line-of-sight or not. We use an artificial neural network and random forest classification for this purpose.

Keywords: self-calibration; localization; ultrasound; machine learning; indoor localization; tdoa; random forest

1. Introduction

The applications for which indoor localization systems are envisioned are immense. Among others, these systems can be used in intralogistics. The domain of in-house logistics of commercial enterprises, sometimes also referred to as intralogistics, covers the entire range of organization, execution and optimization of in-house material flow and warehousing. While traditionally, and by the means of material transportation, a static infrastructure is understood, involving conveyor belts, manually operated cranes and fork lifts, there is a trend towards autonomous agents. An important factor is localization and tracking of these autonomous systems [1,2].

Today, many different indoor localization systems are available based on different methods. A large number of them use radio frequency (RF) signals for localization [3–6]. The RF-Systems use the propagation of radio waves. Another possibility is to use the propagation of sound for localization [7,8]. Indoor localization systems work with indoor satellites (anchor nodes), which have to be installed inside the indoor environment. Most of the traditional indoor localization methods assume that the position of these anchor nodes are known in advance. By knowing their position, the position of the target can be estimated. However, this means that in a real-life scenario, they have to be manually measured, which is a cumbersome job and prone to errors. Therefore, a system which is capable of locating the target without manually measuring the anchor node positions is necessary. We propose two different methods. The first method estimates the positions of the anchor nodes and the target using non-linear optimization. This allows locating the target with an error in the order of centimeters. This method increases the success rate of traditional self-calibration methods by using an inertial measurement unit (IMU) as an additional source of information. Moreover, it is robust against non-line-of-sight measurements, as it discretizes the solution space, choosing the discretized solutions which comply with the information gathered by the IMU.

We propose another method which uses machine learning to identify when the target is in predefined regions. This does not allow localization outside these regions. However, it ensures proper localization in the zones of interest. This is because one can locate the target independently of whether the received signals are in line-of-sight or non-line-of-sight. The algorithm learns from the measurements that are received in the region regardless of this matter. Moreover, the installation effort is reduced even more than estimating the anchor node positions. This is because when estimating the anchor node positions, one needs to ensure that all receivers get enough line-of-sight measurements from different positions in the location area, so that their position can be estimated. Moreover, one needs to ensure that the graph of estimated positions is rigid. This means, there needs to be enough sender positions so that all receivers are constrained by the same coordinate system without allowing any rotation or translation.

In this work, we present a novel method for self-calibration, which determines the anchor positions before the localization of the target. The goal is to improve the success rate of the traditional self-calibration methods. The novel algorithm that is presented in this study simplifies the calibration of acoustic localization systems without compromising the robustness of the system. This is due to the fact that the information gathered from the inertial measurement unit is used to refuse unlikely distributions of receivers.

Additionally, in this manuscript, we show how machine learning algorithms can be used for locating a target when the receiver positions are unknown. This is specially useful in situations where one wants to know whether a speaker is in a certain area. By training the algorithm over different regions, one can then identify in which of the predefined regions the target is located. The results suggest that this method can be applied for zone detection in real-life scenarios.

2. State of the Art

In classic hyperbolic TDoA localization, the positions of static receivers are assumed to be known in order to locate a moving sender. In the scope of this work, we will refer to the stationary nodes as anchors, and as beacons to the localized nodes. In some publications, it is suggested to compensate for given erroneous reference anchor positions [9,10], but in general, the positions of anchors are assumed to be precisely determined by external means in an external coordinate system. Manual measurement of anchor positions may be done by a measuring tape, a laser range finder, a grid arrangement or specifically structured sensor array, or by geodetic methods.

The problem of calibration-free TDoA is challenging. First, the number of unknown variables to be determined is higher than in conventional TDoA, as the anchor positions need to be estimated. Second, the beacon and anchor positions depend on each other, which is adverse to the robustness of the estimation. For instance, one failed measurement at a specific time, a large outlier, affects the estimation of anchor positions, which again affect the estimation of other sender positions. In contrast, in conventional TDoA, a failed measurement can only influence a single sender position. Third, due to the high dimensionality of the problem, many strategies to linearize the problem [11,12] cannot be applied in general, so iterative approaches are required, which are time-consuming and require careful initialization.

For self-calibration, different methods are available. In [13], Biswas and Thrun apply a maximum likelihood estimation algorithm to localize the anchors and the beacon simultaneously. In [14], Wendeberg et al. present a non-linear optimization approach for TDoA self-calibration. In the self-calibration problem, both beacon locations and anchor positions are unknown variables. Starting from the randomized initial values, Wendeberg et al. use gradient descent and the Gaussian–Newton method to search for a minimum of the TDoA objective function.

Due to the non-convexity of the hyperbolic function, there may be multiple local minima. Thus, the non-linear optimization method does not ensure a global minimum. In [14], Wendeberg et al. repeat the randomized initialization to increase the probability of finding the global minimum. Furthermore, in [15], Wendeberg et al. solve the self-calibration problem with a branch-and-bound algorithm. In comparison to the optimization-based method, the branch-and-bound method requires more computational power, as revealed in [16]. In [17,18], a far-field assumption is used to simplify the equations and initialize the variables. In contrast, here we present an initialization technique which utilizes the gyroscope measurements to generate an initial guess close to the true positions. It improves the success rate of self-calibration and increases the converging rate during the optimization process. For the sake of convenience, we name this technique direction adjudgment, or DA in short, and we use DA-SC to refer to the self-calibration initialized by direction adjudgment. Meanwhile, in this work, RA refers to the randomized initialization and RI-SC refers to the self-calibration initialized by randomized initialization.

In this work, we also explore the possibility of using machine learning for detecting whether a speaker is located in some predefined areas, without having to measure or estimate the positions of the anchors. Some authors have explored the possibility of using these algorithms for TDoA localization in the past. For example, Niitsoo et al. [19] use the channel impulse responses for localization. Feig et al. [20] use recurring neural networks on drifting time-of-flight measurements. Zhang et al. [21] use a support vector machine (SVM) for identifying whether an acoustic measurement is in line-of-sight or non-line-of-sight. Ebrahimkhanlou et al. [22] use deep learning to locate acoustic emission sources in plate-like structures.

3. Localization by TDoA with Machine Learning

A simple but effective approach to avoid the tenuous manual measurement of all N_k anchors' three-dimensional positions is to use a machine learning algorithm in order to learn the characteristic TDoAs at each position. This is specially useful in situations where the positional error is less important than a correct association of the position information, e.g., if a customer is at a certain table. Therefore, we define two-dimensional rectangular clusters that mimic this situation. In order to train the algorithms, we put the transmitting beacon in multiple positions $\mathbf{b}$ inside every cluster and save the first reception time t_r of every anchor. We train a neural network and a random forest classifier with TDoA vectors τ that contain all the possible anchor pairing permutations. We train both so that we can compare their performance over different circumstances.

3.1. Time Difference of Arrival (TDoA)

Having one of the N_k anchors with index i located at a three-dimensional position $\mathbf{k}_i$ and a speaker located at a position $\mathbf{b}$, the distance between the two nodes is described in Equation (1). Consequently, a signal traveling from $\mathbf{b}$ to $\mathbf{k}_i$ is received at time t_i as in Equation (2).

$$r_{bi} = \|\mathbf{k}_i - \mathbf{b}\| \tag{1}$$

$$t_i = t_b + \frac{r_{bi}}{c} \tag{2}$$

where c is the sound velocity and t_b the starting time of the signal transmission. Then, having another one of those anchors with index j at position $\mathbf{k}_j$, one can calculate the TDoA τ_{ij} as formulated in Equation (3), which does not depend on the sending time.

$$\tau_{ij} = t_i - t_j = \frac{r_{bi} - r_{bj}}{c}, \qquad i, j \in [1, ..., N_k] \mid i < j \tag{3}$$

In order get a notion of how large the clusters can be, one can study what is the minimum achievable position error using TDoA and assuming every timestamp has additive Gaussian noise with standard deviation σ_n.

When using hyperbolic multilateration, two timestamps are used for every measurement, which means the noises between multiple measurements are correlated. As shown in [23], one can define a matrix $\mathbf{D}$ as:

$$\mathbf{D} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{4}$$

The first column corresponds to the reference anchor whose timestamp is subtracted to the others. The other $N_k - 1$ columns correspond to the other anchors. Then the $(N_k - 1) \times (N_k - 1)$ noise matrix of the TDoA measurements will be:

$$\mathbf{R}_{\text{TD}} = \sigma_n^2 \mathbf{D}\mathbf{D}^T \tag{5}$$

Then, the matrix $\mathbf{A}$ which contains an estimation of the covariance of the estimated variables can be calculated as follows:

$$\mathbf{A} = (\mathbf{H}^T \mathbf{R}_{\text{TD}}^{-1} \mathbf{H})^{-1} \tag{6}$$

where $\mathbf{H}$ is the Jacobian matrix of the TDoA sensor model. The squared root of the trace of $\mathbf{A}$ is a lower bound for the achievable root mean squared error assuming a non-biased estimator:

$$\text{RMSE}(\widehat{\mathbf{b}}) \geq \sqrt{\text{Tr}\left(\mathbf{A}\right)} \tag{7}$$

where $\widehat{\mathbf{b}}$ is the estimation of $\mathbf{b}$. Moreover, if this term is normalized by the noise, it is known as the dilution of precision and is used to determine how the geometrical distribution of anchors affects the final position estimation:

$$\text{DOP} = \sigma_n \sqrt{\text{Tr}\left(\mathbf{A}\right)} \tag{8}$$

3.2. Neural Network Classification

An artificial neural network consists of a high number of neurons n which are interconnected over a certain number M of layers. The layers between the input and the output layer are the hidden layers and determine the complexity of the functions which they approximate (see Figures 1 and 2). Then, the connection between a neuron i of the layer $k - 1$ with another neuron j of the next layer k is weighted with a weight $w_{i,j}$ before being added to the other weighted inputs of the neuron. Afterwards, each neuron adds a bias $b_{j,k}$. The output $o_{j,k}$ of the neuron j in the layer k is calculated by the activation function $\text{g}(\cdot)$. This can be written as follows [19]:

$$o_{j,(k+1)} = \text{g}(b_{j,k} + \sum_{i=0}^{N} w_{i,j} n_{i,k}) \tag{9}$$

where N is the number of neurons per layer. The value of $n_{i,k}$ is the activation of a neuron in the previous layer.

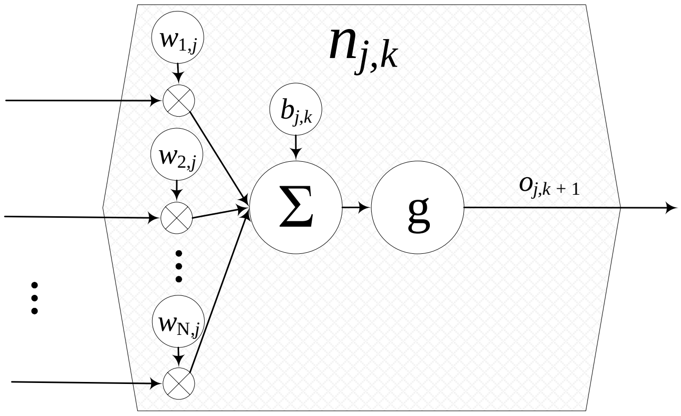


Figure 1. Schematic of a neuron, that can be joined with others to form an artificial neural network.

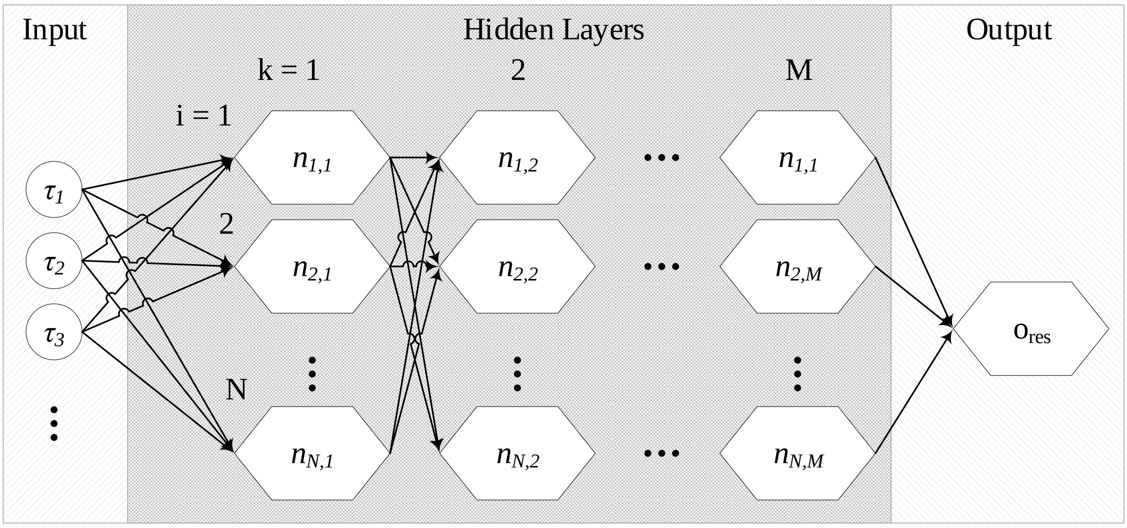


Figure 2. Schematic of an artificial neural network.

3.3. Random Forest Classification

The random forest consists of multiple trained trees. These trees have multiple splits where the data gets separated into two similar sized groups. Possible criteria for the split are the gini index or the entropy. See Figure 3 for a simple example with five clusters C_1 to C_5. In each branch, we have a simple condition based on one TDoA measurement. If one would use only one tree, a problem would arise. If we do not have any data about the root node or any node along the way, the tree cannot make a prediction, or at least, it can only be used to analyse a subset of clusters. To work around this problem, we do not train one tree but a forest of trees. All trees are trained on different subsets of the training data and use different splits to distinguish between the clusters. As well as reducing our reliance on each TDoA measurement, the ensemble of trees also reduces the bias that a single trained tree has. Another advantage of the random forest is that it is very easy to interpret as we can look into the splitting points and see how the clusters get separated.

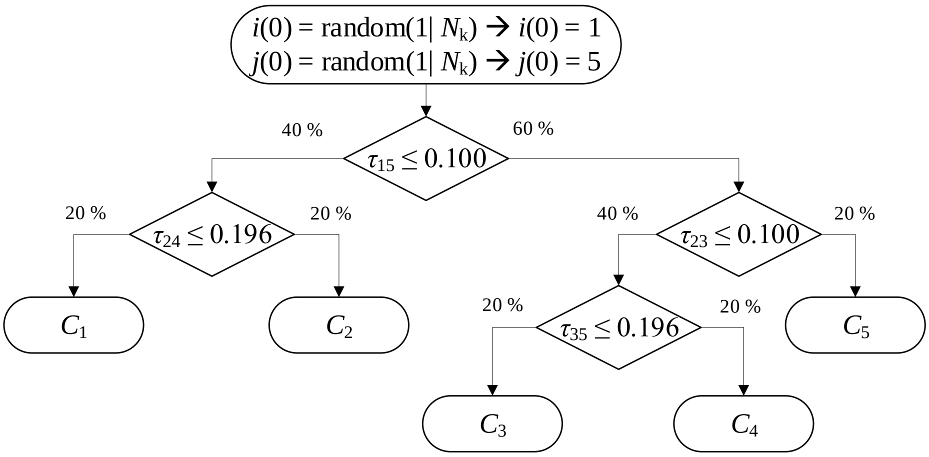


Figure 3. Simple example of a decision tree with 5 clusters. Depending on the TDoA measurements a cluster can be chosen.

4. IMU-Assisted Self-Calibration Algorithm

In this case, we aim to estimate the unknown variables: the position of the beacon $\mathbf{b}$ and the position of the anchors $\mathbf{k}_i$. In order to avoid that the non-linear optimization algorithms get stuck in local minima, we first estimate a suitable initial value for these variables by additionally measuring the rotational speeds with a gyroscope; in our case, embedded in an IMU. While the additional sensor generally requires a separate calibration on its own and increases the number of error sources, the proposed method only uses the gyroscope data for a rough estimation that is less influenced by the sensor error. Further signal pre-conditioning of the gyroscope data can be implemented to increase this estimation accuracy and limit the search interval for each anchor even further [24].

The proposed initialization algorithm assumes that at the beginning, the beacon is located under an anchor. Consequently, one can estimate the distance to other three anchors. By knowing these distances, one can define the coordinate system in such a way that two anchors are fixed and the other two must lie each one in a different circle. The solutions inside such circles are discretized into equally spaced points. When the target moves, it generates measurements which must be consistent with the estimated positions. The measurements from the anchors and the ones from the gyroscope are used to decide the most likely solutions inside the discretized solution space. Once the positions of the first four anchors have been estimated, the other anchors can also be located using the estimated sender positions. The details of this algorithm are provided in the next sections.

4.1. Anchor Distances and Possible Space Reduction

The unknown anchors are assumed to be located on two-dimensional planes with known fixed heights. Hence, the possible space of each anchor is a horizontal plane which has two degrees of freedom. If we take one anchor at position $\mathbf{k}_i$ as the pivotal reference point and estimate the horizontal distance to another encompassing anchor at position $\mathbf{k}_j$, we can reduce the possible space of the latter to a circle whose midpoint is $\mathbf{k}_i$ and its radius r_{ij} equals to the geometric distance between the two positions, as formulated in Equation (10). Note that r_{ij} is the projection of the geometric distance onto the lateral plane in the height of the beacon, in contrast to the distance between beacon and anchors (compare Equation (1)). This is also illustrated in Figure 4.

$$r_{ij} = \left\|\mathbf{k}_j - \mathbf{k}_i\right\| \tag{10}$$

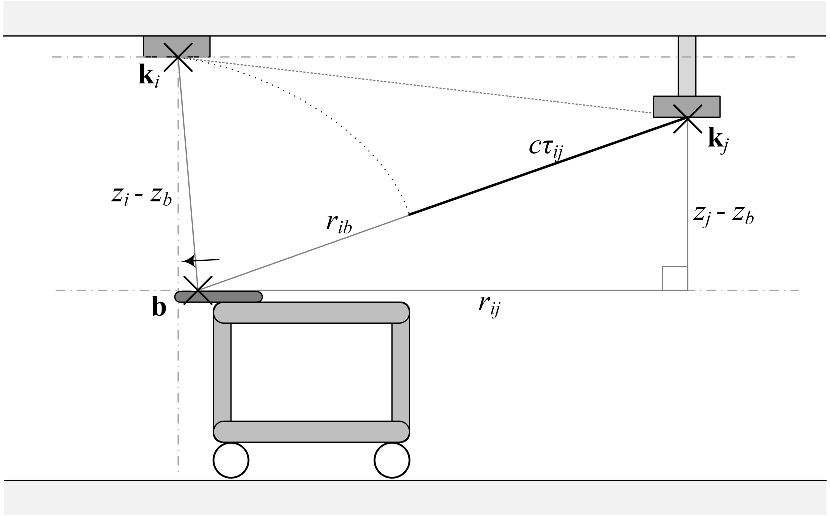


Figure 4. Distance estimation set-up, that reduces the degrees of freedom.

In order to estimate the distance on the two-dimensional plane between those two anchor positions, we place the sender vertically below the anchor at the reference position. This allows for a simplified calculation of the time difference of arrival between the reference anchor and encompassing anchor.

$$r_{ij} = \sqrt{\left(r_{bj}\right)^2 - (r_{bi})^2} \tag{11}$$

$$r_{bi} = \sqrt{(x_i - x_b)^2 + (y_i - y_b)^2 + (z_i - z_b)^2} \tag{12}$$

$$r_{bj} = r_{bi} + c\,\overline{\tau}_{ij} \tag{13}$$

Let $\overline{\tau}_{ij}$ be the mean value of the TDoA measurements and z_i, z_j and z_b be the heights, we can regard the distance r_{bi} in this specific arrangement as a entirely vertical and one-dimensional problem, as shown in Equation (14).

$$\lim_{x_b \to x_i, y_b \to y_i} r_{bi} = z_i - z_b \tag{14}$$

$$\lim_{x_b \to x_i, y_b \to y_i} r_{bj} = z_i - z_b + c\,\overline{\tau}_{ij} \tag{15}$$

and thus we can express the horizontal distance r_{ij} (see Figure 5) as the geometric distance formulated in Equation (16).

$$r_{ij} = \sqrt{\left(z_i - z_b + c\,\overline{\tau}_{ij}\right)^2 - (z_i - z_b)^2}. \tag{16}$$

As the anchors are on the ceiling and our vertical coordinate increases from floor to ceiling, we assume that $z_i \approx z_j \gg z_b$ always holds.

Equations (14) and (16) are only valid when the condition that the beacon is right under the pivotal anchor holds. We observed with a Motion Capture system [25] that the placement error due to our visual estimation remains approximately below 0.3 m to the actual position of the anchors. It is important to note that the error of human observation in this phase only influences the initialization of the variables. The aim is to find an initial value for the variables of the scenario which can then be refined with nonlinear optimization without getting stuck in local minima.

As mentioned before, the possible space of the anchor can be reduced to a circle if its distance to the pivotal anchor $r_{1,i}$ is known. We further discretize the circle as discontinuous points on the circle. As the encompass anchor is at the coordinate origin in the x and y coordinates, the discretized positions of the pivotal anchors $\mathbf{k}_i^-$ are defined in Equation (17).

$$\mathbf{k}_i^- = \begin{pmatrix} r_{1,i}\cos(\phi_{i,k}) \\ r_{1,i}\sin(\phi_{i,k}) \\ z_i \end{pmatrix}, \tag{17}$$

where z_i is the known height of the anchor, r_{1i} is the estimated inter-anchor distance and $-\pi \leqslant \phi_{i,k} < \pi$. We discretize the angle as steps of $\Delta\phi$:

$$\phi_{i,k} = k\,\Delta\phi \tag{18}$$

where $k \in \mathbb{Z}$. Here $\Delta\phi$ decides the resolution of the discretized approximation.

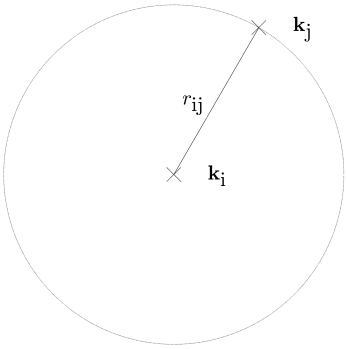


Figure 5. Knowing the distance between the pivotal anchor and the encompass anchor, the pivotal anchor must lie in a circle whose center is the encompass anchor.

In order to reduce the computational complexity, we avoid the congruent transformation of anchor configurations. We establish a coordinate system where the pivotal anchor is fixed at the origin and one of the encompass anchors on the positive half of the x-axis, i.e., at the point $\mathbf{k}_2 = [r_{12}, 0]$, where r_{12} is the estimated distance between the pivotal anchor $\mathbf{k}_1$ and the first encompassing anchor $\mathbf{k}_2$. The possible spaces of the other two anchors $\mathbf{k}_3$ and $\mathbf{k}_4$ are circles around the pivotal anchor. Figure 6 gives an example for the first four anchors.

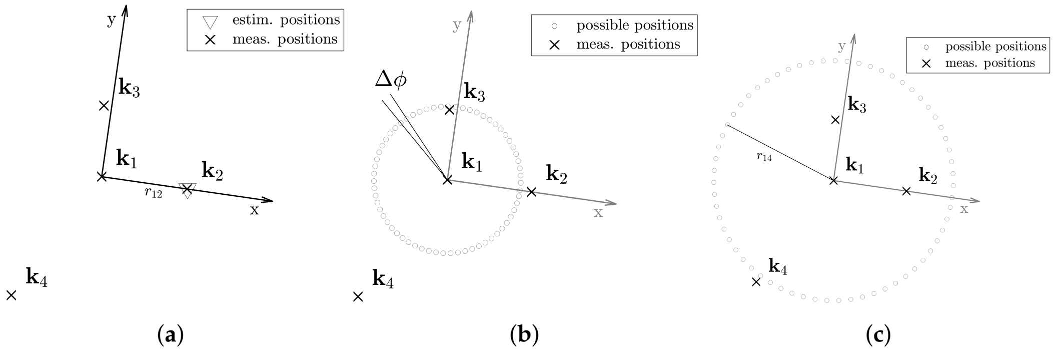


Figure 6. Reduced possible spaces for four anchors. In (**a**) we establish a new coordinate system. The pivotal anchor $\mathbf{k}_1$ is fixed at the origin in x and y. The estimation of the anchor $\mathbf{k}_2$ is fixed at $x = r_{12}$, $y = 0$. The possible space of the additional anchors $\mathbf{k}_3$ in (**b**) and $\mathbf{k}_4$ in (**c**) are on circles around the origin.

4.2. Straight Movement Detection

In order to further reduce the subspace of possible solutions $\mathbf{k}^-$, we use combinations of four anchors from the subspace and estimate the sender positions. Afterwards, we compare the results with the measurements from the inertial measurement unit and discard the anchor positions which lead to sender positions which contradict the IMU estimations. In order to do so, we use the inertial measurement unit to detect when the target is moving in a straight line. The heading direction remains unchanged when this happens. Therefore, the horizontal rotation is close to zero. We first smooth the rotation measurement ω_z and then apply a threshold on the smoothed curve. The endpoints of the straight-moving intervals can be found around the intersections of the threshold line and the smoothed ω_z curve.

An example is given in Figure 7. We can observe several peaks and saddles in the ω_z curve. The peaks occur when the vehicle turns its direction and the saddles indicate that the beacon is moving in a straight direction. The magenta points denote the endpoints of the straight-moving interval, while the green points are the midpoints of the time interval. In Figure 8, we can see how the straight lines are successfully detected in a real trajectory. We name the endpoints of the time interval in which the target moves in a line as the *critical time points*, their location are the *critical waypoints* and the corresponding TDoA measurements the *critical TDoA sequences*. In this work, a *waypoint* refers to a location sample on the trajectory. Each waypoint is associated with a TDoA sequence.

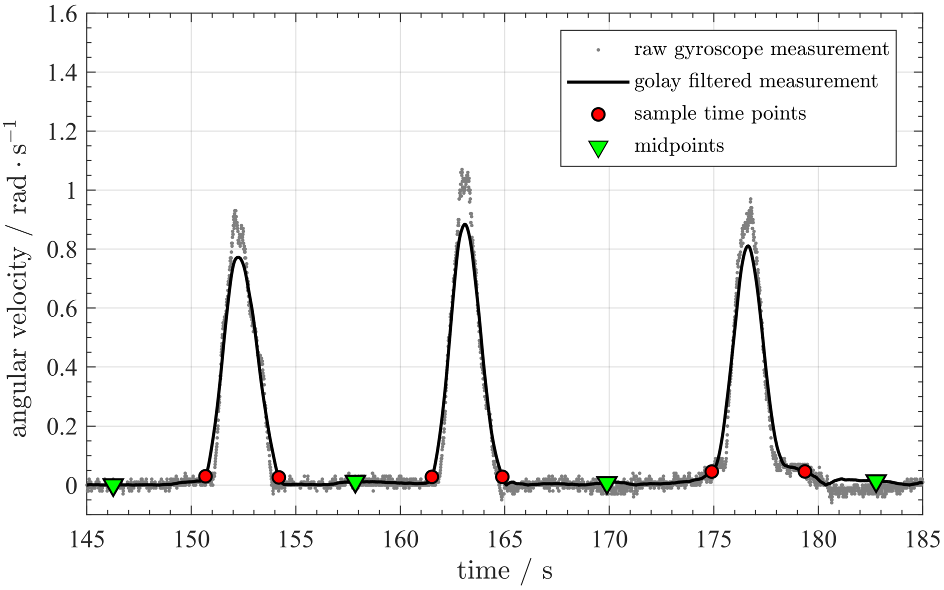


Figure 7. Extracting the time interval when a beacon moves in straight lines. The magenta points denote the sample points where the straight moving interval starts and ends. The green points are the midpoints of time in the straight moving intervals.

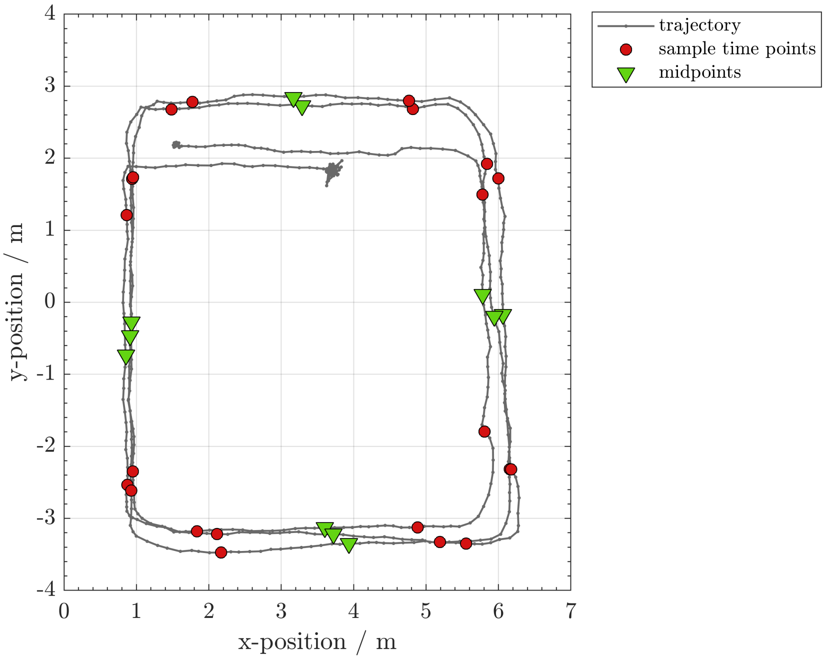


Figure 8. Using the inertial measurement unit, one can detect the straight lines. The shown trajectory is estimated using ultrasound localization and known anchor positions. While this is not the case, it shows how the target moved during the calibration phase. The magenta points are critical points which define the starting and ending points of the straight lines.

4.3. Direction Adjudgment

In order to reduce the computational complexity and the calibration effort, we first assume that we have estimated the distance from the encompass anchor to three pivotal anchors. Then, we have four anchors, which are enough to estimate the position of the target using TDoA. Two of these anchors will have a fixed position and two of them will be located in circles; as it can be seen in Figure 6. In order to find which are the most likely positions of the anchors inside the circles, we use the IMU data. For every possible position of the anchors, we estimate the positions of the target using TDoA. In order to do this, we use a closed-form [26] formulation. From the TDoA estimations, we extract the forward directions and compare them with the rotations measured by the gyroscope. The similarity between the trajectory measured by the IMU and the one estimated with the TDoA measurements lets us know which are the most likely anchor positions.

If a target turns, then moves straight and then turns, the line is defined by the critical points $\mathbf{b}_j^h$,$\mathbf{b}_j^e$ which denote the starting and ending points of the line. Then, we can estimate the direction $\hat{\theta}_j$ for the straight segment j with

$$\hat{\theta}_j = \arctan(\widehat{\mathbf{b}}_j^e - \widehat{\mathbf{b}}_j^h) \tag{19}$$

where $\hat{\theta}_j$ is the estimated direction of the straight segment j, $\widehat{\mathbf{b}}_j^h$ is the estimation of the start critical point and $\widehat{\mathbf{b}}_j^e$ is the estimation of the end critical point.

Rotation Calculation

In order to estimate the rotation, we integrate the horizontal rotation rate in the time interval $[t_{m1}, t_{m2}]$. We formulate it in a generalized time interval $[t_{m,j}, t_{m,j+1}]$ as

$$\Delta\theta_j = \int_{t_{m,j}}^{t_{m,j+1}} \omega_z(t)\, dt, \tag{20}$$

where $\Delta\theta_j$ is the measured rotation between the straight segments j and $j+1$, and ω_z is the rotation rate around z-axis of the vehicle frame.

In the end, we apply

$$\left|\theta_{j+1} - \theta_j - \Delta\theta_j\right| < \theta_{th} \tag{21}$$

to judge the direction, where θ_{th} is a chosen threshold. If the measurements fulfill Equation (21), we consider the guessed anchor positions to be feasible.

The trajectory may consist of several straight segments. Each pair of adjacent straight segments forms a constraint of the form formulated in (21). We sum up the number of satisfied constrains for each of the guessed configurations. In the end, we keep only the guesses with the largest number of satisfied constrains and exclude the others. We provide an example of the reduced possible spaces after direction adjustment in Figure 9. Note that the arc-shaped possible space is still not the initialization. We average the remaining φ_k^i of an anchor i and initialize it as $\mathbf{k}^-$, where $\bar{\varphi}_k^i$ is the mean of the feasible angles. We provide an example of the result in Figure 9. The hollow circles are the remaining anchor positions while the points used for initialization are denoted as filled circles in Figure 9.

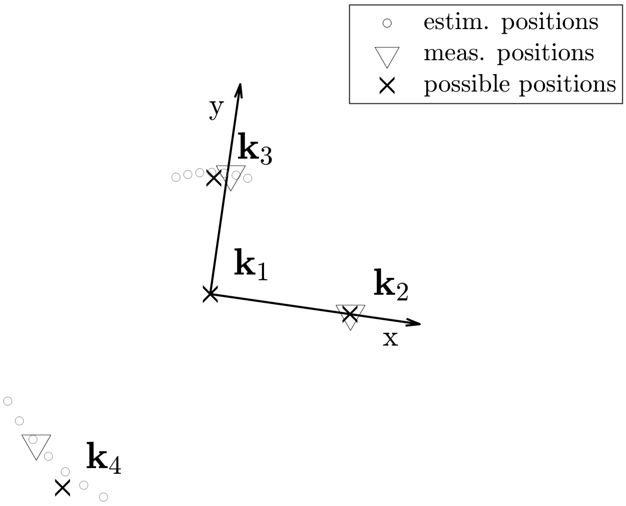


Figure 9. Possible space and initialization after direction adjudgment . Notice that the subspace of possible solutions is reduced from circles to arcs and then to points $\mathbf{k}_3$ and $\mathbf{k}_4$, which are used for initialization.

In this section, we use the closed-form TDoA solution presented by Bucher and Misra in [26] to estimate the target positions. This closed-form solution requires four anchors. When the number of anchors increases, the possible solutions increase exponentially, which makes the direction adjudgment inefficient. Therefore, we introduce a new method to initialize the additional anchors.

4.4. Initialization of Additional Anchors

With the first four anchors initialized, we can use the initialized positions to approximate the true positions of the first four anchors. The subspaces of possible positions for the additional anchors are also circles. First, we randomly select several TDoA sequences and estimate the corresponding sender positions with the approximation of the first four anchors. With the estimated sender positions, we reconstruct the TDoA measurements that each additional anchor would receive and compare them with the measured TDoAs. Repeating this process over all the possible positions of the additional anchors, we choose the possible positions whose resulting forward directions comply best with the gyroscope measurements for initialization.

To initialize the trajectory, we randomly choose a configuration from the possible spaces for each of the waypoints on the trajectory and estimate the sender positions with the closed-form solution.

Figure 10 gives an example of the initialization results. We choose a central anchor $\mathbf{k}_1$ as the origin and a randomly selected encompassing anchor $\mathbf{k}_2$, through which we define the positive x-axis. The position estimation of the next anchors $\mathbf{k}_3$ and $\mathbf{k}_4$ is selected from the reduced solution sub-space. We estimate the initial positions of these latter two anchors from the average coordinates in the polar system.

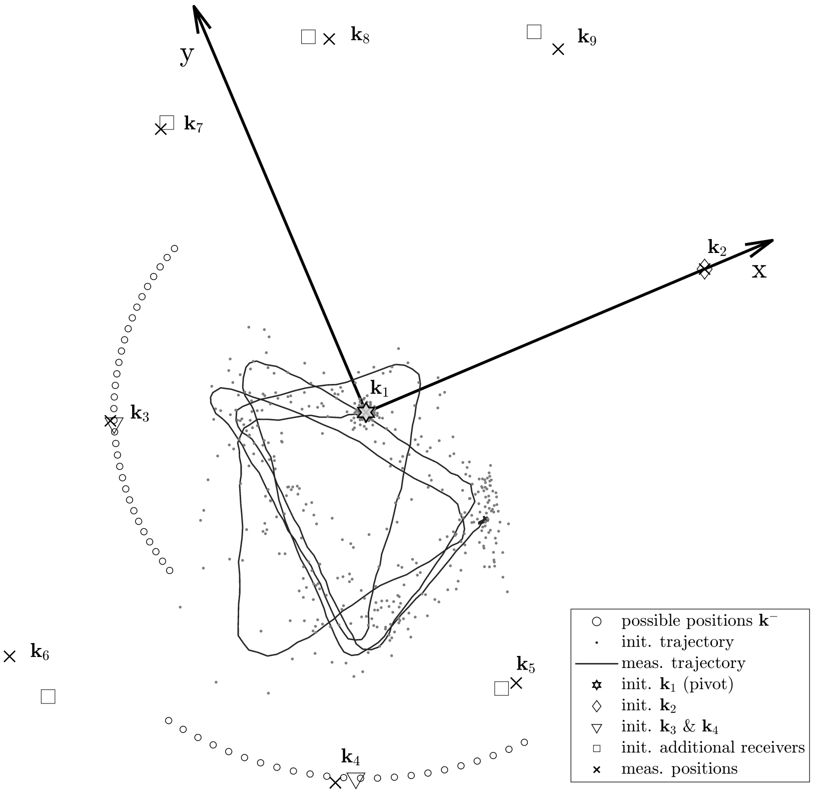


Figure 10. Initialization with direction adjudgement example.

4.5. Optimization

Once we have an initialization for the position of the anchor nodes and trajectory of the sender we minimize the following quadratic error function:

$$E_{ij}^{\text{TDoA}} = r_{bi} - r_{b1} - c\,\tau_{1i}. \tag{22}$$

The objective function is the squared sum of TDoA residuals, which is formulated as

$$\underset{\mathbf{k_i},\mathbf{b_j}}{\arg\min} \sum_{i=2}^{N_k} \sum_{j=1}^{m} \left(E_{ij}^{\text{TDoA}}\right)^2. \tag{23}$$

In order to search for the minimum value of the objective function, we apply the Levenberg-Marquart algorithm.

5. Simulations

We designed and performed a simulation in order to verify that the direction adjudgment can be generalized to different anchor topologies.

The simulation depends on both the IMU and TDoA inputs. However, we are not able to simulate the IMU measurements since the theoretical models often differ highly from reality. Therefore, we use real IMU measurements. Based on the randomly generated anchor positions and the recorded beacon trajectories, we estimate the theoretical values of TDoA measurements and impose Gaussian distributed errors on the theoretical values. The assumption of a Gaussian distribution does not hold, as was also shown by Cui et. al. [27], but for this work we estimated the resulting error as small enough to be neglected at this stage.

5.1. System Error Distribution

First, we have to determine the error level of the TDOA localization system before starting the simulation. Therefore, we estimate the theoretically expected TDoA values using the real trajectories recorded by a Motion Capture system, which has submillimeter accuracy and precision, and the true anchor positions. We compared the TDoA measurements and the expected values in 200,138 TDoA measurements to estimate the measurement errors (see Figure 11). Fitting the error distribution into the Gaussian distribution, we obtained a Gaussian distribution with $\mu_{\text{TDoA}} = -0.006\,\text{ms}$ and $\sigma_{\text{TDoA}} = 0.637\,\text{ms}$. Since $\mu \ll \sigma$, we regard the offset of the error distribution to be 0.

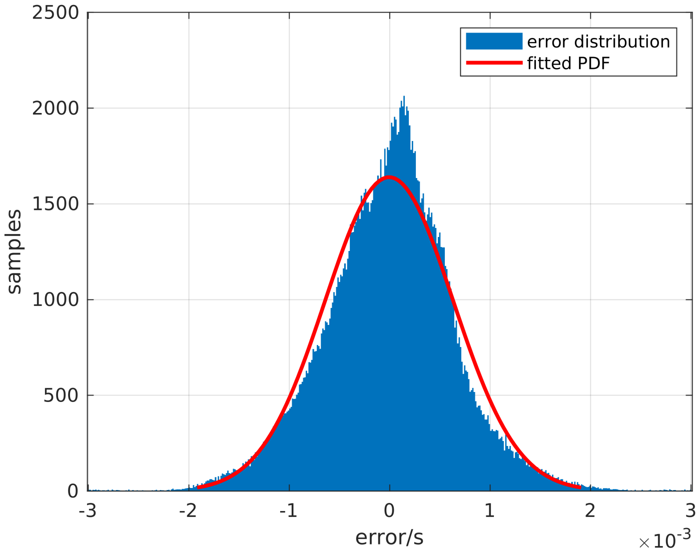


Figure 11. Fitting the TDoA error with a Gaussian distribution.

Actually, the localization system measures the timestamp of the signal of arrival. The TDoA is the subtraction of two timestamps measured by the different anchors. Assuming that the errors of the timestamp measurements from all the anchors comply with the same Gaussian distribution, i.e., $e_t \sim \mathcal{N}(\mu_t, \sigma_t^2)$, we have:

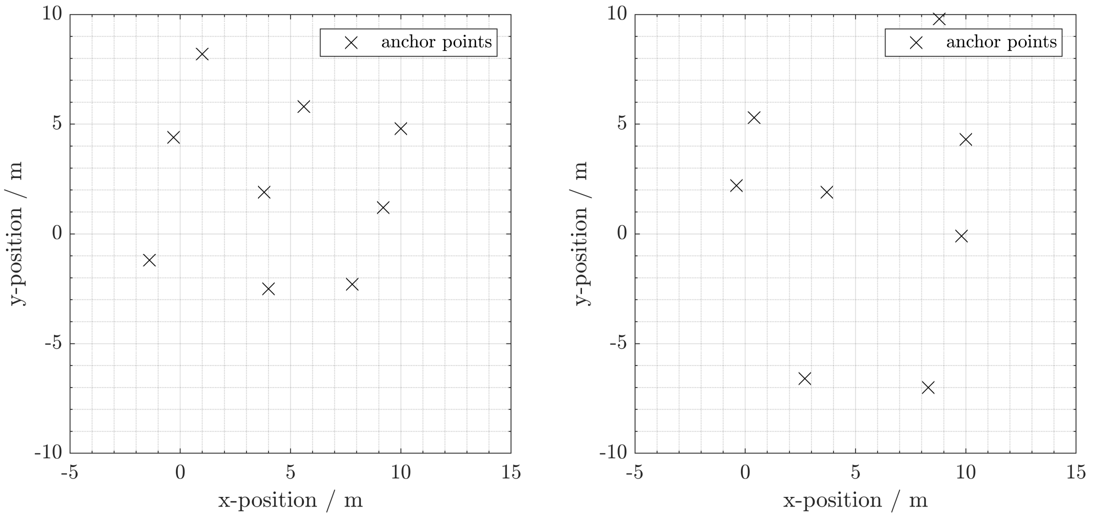
$$e_{\text{TDoA}} = e_{t,i} - e_{t,1} \sim \mathcal{N}(\mu_t, \sigma_t^2) + \mathcal{N}(-\mu_t, \sigma_t^2) = \mathcal{N}(0, 2\sigma_t^2). \tag{24}$$

Therefore, the standard deviation is calculated as:

$$\sigma_t = \sqrt{\frac{1}{2}\,\sigma_{\text{TDoA}}^2} = 0.45\,\text{ms}. \tag{25}$$

5.2. Performance Comparison of DA vs. RI

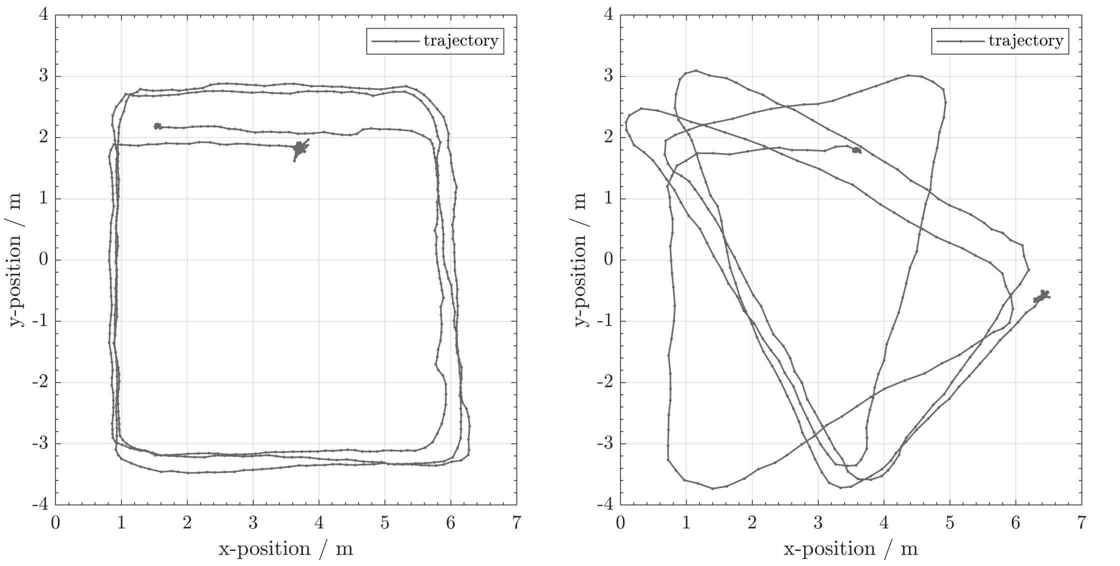
We randomly generated 1000 different anchor configurations. Each configuration contains nine anchors. In reality, one would not install two anchors very close to each other. Therefore, we set the minimum distance between two arbitrary anchors to be three meters. Figure 12 gives examples of simulated anchor configurations. In the simulation, we use two target trajectories (see Figure 13). Since we generate 1000 anchor configurations, we will have 2000 different simulation scenarios. In the end, we compare the performance between DA-SC and RI-SC over all the scenarios.



(**a**) First example of an anchor configuration.

(**b**) Second example of an anchor configuration.

Figure 12. Two examples of the 2000 simulated anchor configurations.



(**a**) First trajectory used in the simulations.

(**b**) Second trajectory used in the simulations.

Figure 13. The two trajectories shown in (**a**) and (**b**) respectively are used in the simulations to emulate a moving beacon.

One of the performance indicators is the success rate. In the self-calibration problem, the anchor positioning error larger than 0.5 m is mainly due to local minima. Therefore, we have the following definition of success:

Definition 1. *Success of Self-calibration: A self-calibration attempt succeeds if the following condition holds*

$$\max\Big(\{\|\widehat{\mathbf{k}}_i - \mathbf{k}_i\|\}\Big) < 0.5\,m \tag{26}$$

where $\widehat{\mathbf{k}}_i$ is the estimation of the anchor position and $\mathbf{k}_i$ denotes the true position.

The direct self-calibration results are likely to be in a different coordinate system to the true anchor positions. The estimations are rotated and translated according to the reference system before the comparison.

Since direction adjudgment is a deterministic process, we evaluate only its single success rate from the first attempt. Table 1 gives the success rate at the first attempt between direction adjudgment (DA-SC) and random initialization (RI-SC) for the two different trajectories.

Table 1. Comparison between success rates at the first attempt of the two self-calibration methods Direction Adjudgment (DA-SC) and Random Initialization (RI-SC).

	Success Rate/%	
Trajectory	**DA-SC**	**RI-SC**
1	98.9	41.8
2	100	50.9

In addition, Figure 14 shows the difference of performance between DA-SC and RI-SC. The corresponding success rates given in Figure 14 and Table 1 are the percentage of simulated topologies that were successfully estimated by DA-SC. In contrast, randomized initialization is a stochastic process. We repeat the RI-SC for 20 times redoing the unsuccessful cases and count the accumulative success rate after each repetition. We observe that the success rates of DA-SC are nearly 100%. Comparatively, the success rates of RI-SC can only reach this level after 10–20 repetitions.

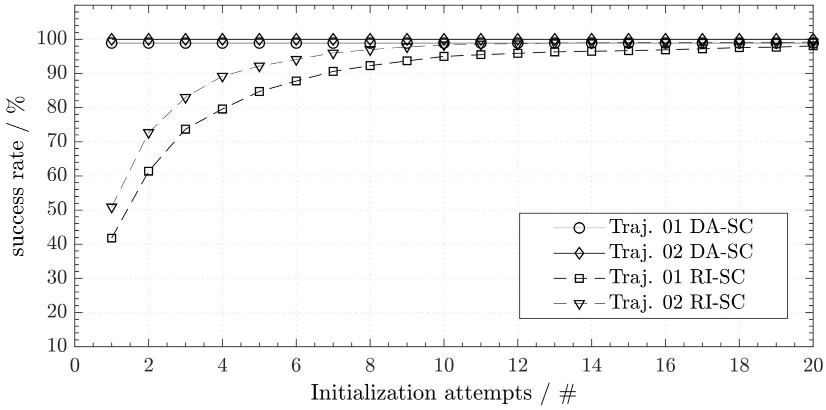
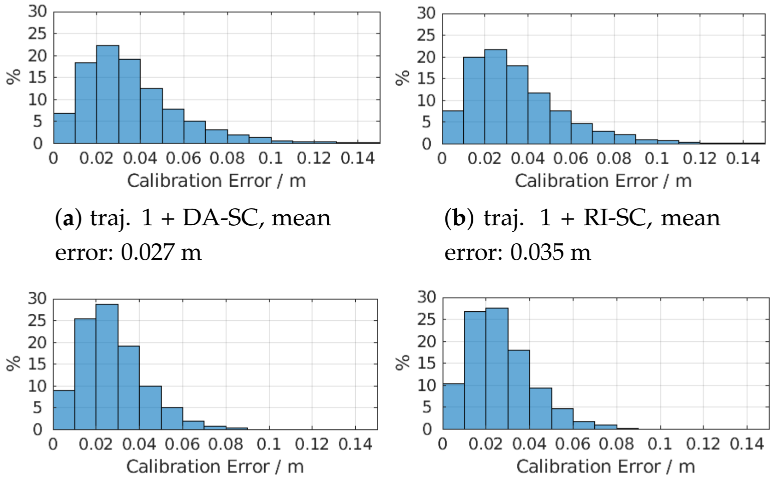


Figure 14. Comparison between the success rate of different initialization methods.

As shown in Figure 15, since we use the same objective function, the calibration results have a similar error and the resulting mean errors have nearly the same value.



(**a**) traj. 1 + DA-SC, mean error: 0.027 m

(**b**) traj. 1 + RI-SC, mean error: 0.035 m

(**c**) traj. 2 + DA-SC, mean error: 0.026 m

(**d**) traj. 2 + RI-SC, mean error: 0.035 m

Figure 15. Error distributions for different simulation configurations.

6. Experiments of IMU-Assisted Self-Calibration

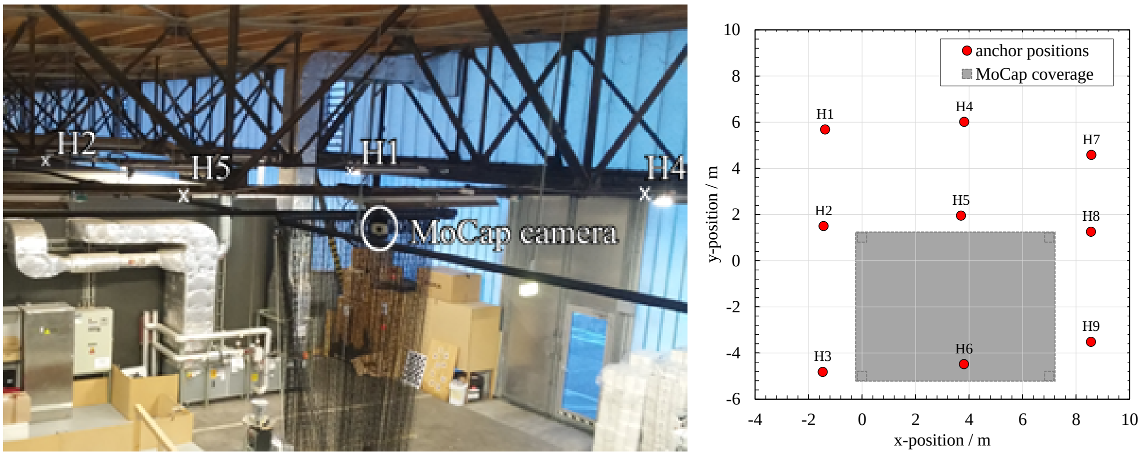
We performed experiments for the verification and further analysis of the simulation results.

6.1. Locales and Set-Up

We present measurements from the two seperate locales *Hangar* (see Section 6.1.1) and *Messe* (see Section 6.1.2). These installations have different characteristics and, therefore, allow us to test our proposed algorithm under different conditions. The receivers are placed at a larger height in the Messe, which increases the dilution of precision. However, in the experiment in Hangar there are more obstacles and, therefore, the signals can be reflected or occluded.

6.1.1. Experiments in the Hangar

The anchors in the Hangar are mounted in a height of appoximately $z_H \approx 4.9\,\text{m}$ above the concrete floor to steel beam girders and powered permanently by wire. We measured the real positions of the anchors with a Total Station *Topcon GPT-8203A* for reference. A part of the anchor set-up is shown in Figure 16a for illustration. We use the Motion Capture (MoCap) system *MotionAnalysis Cortex* with *Raptor-E* cameras to record the true trajectory with sub-millimeter precision. The MoCap system only provides reliable positions in the area between the cameras, which results in a limited coverage area that is considerably smaller than the one of ASSIST. Figure 16b is the top-view of the Hangar experimental area with anchor positions, where the grey area indicates the MoCap coverage.

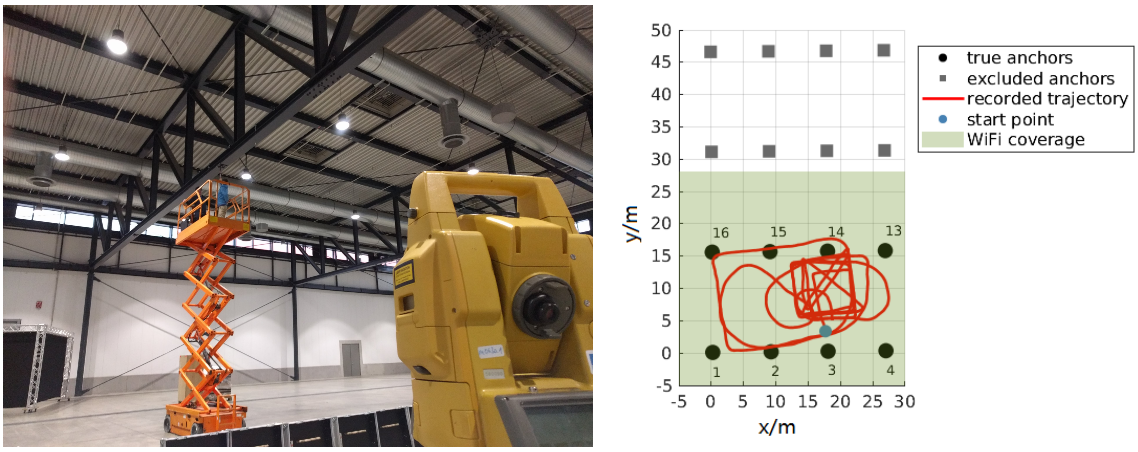


(**a**) (**b**)

Figure 16. The positions of the anchors H1-9 in the Hangar experiments with an exemplary trajectory of experiment 09. There are 12 anchors in hangar of which we use only 9. The shaded area indicates the coverage of the MoCap system. (**a**) Photo of the Hangar ceiling installation. (**b**) Anchor positions and MoCap coverage.

6.1.2. Experiments in the Exhibition Hall

We installed 16 anchors in the commercial exhibition hall “Halle 3” of Messe Freiburg at the bottom of the steel girders at a height of $z \approx 7.3\,\text{m}$, as shown in Figure 17a. The smartphone requires Wi-Fi connectivity to synchronize the IMU and ASSIST. Due to the limited Wi-Fi coverage, we can only calibrate eight of the anchors. Figure 17b visualizes the anchor positions and an exemplary recorded trajectory from our experiment.

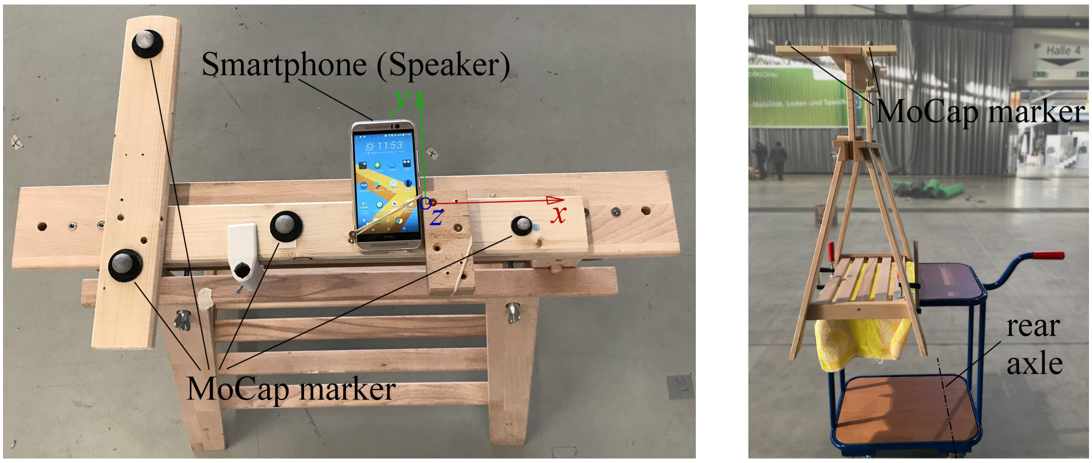


(**a**) (**b**)

Figure 17. The anchor positions in the exhibition hall 3 of Messe Freiburg with an exemplary trajectory of Exp. 16. Notice the error of about 4 m in the start point location: The true start point of the trajectory should be closer to anchor M03 as the beacon was placed directly below this anchor. (**a**) Photo of the anchor node installation at the Messe. (**b**) Anchor positions and Exp. 16 trajectory.

6.2. Experimental Vehicle

We use a *HTC One* smartphone for both ASSIST and IMU tracking. The smartphone is fixed to a wooden support rack along with the motion capturing (MoCap) markers, and this is shown in Figure 18a. We align the smartphone to the midpoint between the rear wheels with the y-axis of IMU perpendicular to the rear axle. Moreover, we fix it horizontally with its screen facing upwards. As a result, the speaker, which is located above the top of the device's screen, is suspended in air. This is done to roughly mimic the smartphone being held in front of your chest while moving and to reduce offset errors. This set-up is then fixed to a push-cart as shown in Figure 18b, which we slowly moved through the experimental areas. The cart has two freely rotating front wheels and two fixed-heading rear wheels, known as an Ackermann steering geometry, which is common for wheeled vehicles [28]. All the wheels are nonholonomic which prevent the side slipping. The heading direction of the vehicle's body is always perpendicular to the rear axle [28], i.e., the moving direction is perpendicular to the rear axle.



(**a**) Smartphone and MoCap markers placement on the wooden rack.

(**b**) Push-cart used in the experiments.

Figure 18. The smartphone on the wooden rack is laterally aligned to the midpoint of the rear axle, with the x-axis of the smartphone in parallel to the rear axle.

6.3. Experimental Results

In the anchor self-calibration experiments in the Hangar we calibrate nine anchors. We also validate the direction adjudgment with 17 real-world experiments. Table 2 summarizes the 17 experiments, among which Exp. 01–12 were conducted in Hangar and Exp. 13–17 were in Messe Freiburg. We applied both direction adjudgment self-calibration (DA-SC) and randomized initialization self-calibration (RI-SC) over all experiments. We repeated the RI-SC for 100 times in each experiment, but execute the DA-SC for only once. Table 2 gives the comparison between DA-SC and RI-SC. We list the results of both initialization and optimization for DA-SC. To evaluate the success rate of RI-SC, we repeat the RI-SC for 100 times and calculate the percentage of success attempts. Note that we use Definition 1 again in this section to define the success of self-calibration. The result shows how the success rate is increased while the mean error is similar in both cases, as expected from the simulated environment. Of special interest are the experiments in the Hangar, as their success rate is limited using randomized initialization due to the large height of the receivers, which increases the dilution of precision. It is worth mentioning that the mean error in these cases is calculated with a reduced number of valid attempts. This is the reason why one observes a larger discrepancy in the mean error of both initialization strategies.

Table 2. Performance comparison between DA-SC and RI-SC in real experiments. The success rates of RI-SC is the percentage of success attempts over 100 repetitions.

	DA-SC	**RI-SC**	
Experiment	**Mean Error/m**	**Success Rate**	**Mean Error/m**
01	0.12	57%	0.13
02	0.12	80%	0.11
03	-	*error*	-
04	0.13	17%	0.14
05	0.10	25%	0.11
06	0.06	90%	0.07
07	0.15	23%	0.15
08	0.09	85%	0.09
09	0.13	67%	0.11
10	0.13	84%	0.10
11	0.08	69%	0.09
12	0.12	67%	0.11
Overall	0.11	60%	0.11
13	0.28	5%	0.30
14	0.10	1%	0.36
15	0.20	0%	-
16	0.10	0%	-
17	0.19	0%	-
Overall	0.17	1%	0.33

7. Experiments of Zone Detection with Machine Learning

The aim of this experiment is to compare the performance of each algorithm in whether it correctly associates the presence of beacons with the correct zone.

7.1. Experimental Setup

In order to test the capability of an artificial neural network to locate the target with unknown anchor positions and a realistic environment, we perform a real experiment in a cafeteria (see Figure 19). We mount seven anchors on the ceiling. Then, we place a sender in 29 different zones on tables, that are marked by tape. The received timestamps at each zone and the zone number serve as the training set for the machine learning algorithms.

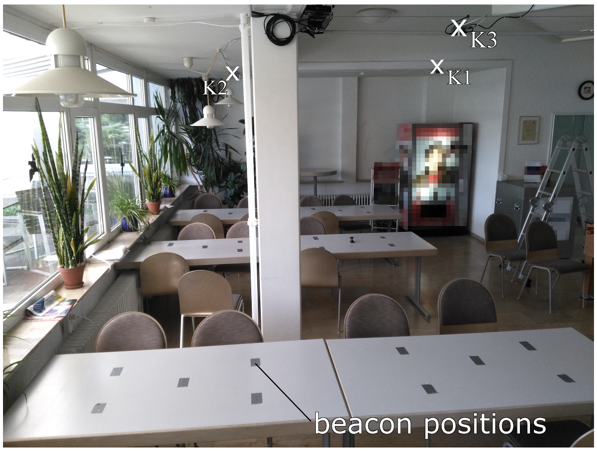


Figure 19. Experimental scenario for the machine learning experiments. A beacon is put on a table approximately at the marked cluster positions and the aim is to correctly estimate the cluster it is in by training each machine learning algorithm with the TDoA measurements of multiple positions on the tables.

7.2. Test Set Creation

We create a test set by placing the sender near each of the trained points in order to estimate the percentage of correct associations. We move the sender near every trained point so that we can simulate a real environment, where the sender is not placed exactly in the trained points.

7.3. Experimental Results

After evaluating the data, there were 91% correct associations using random forest and 88% using a neural network. Note that most of the cluster errors occur in close geometric proximity, and consequently similar TDoA. This is indicated by the arrows in Figures 20 and 21. If one allows a certain geometrical error in the estimations, the number of correct assignments is higher. For example, if only the correct table is to be identified, but not the individual position at the table, only the larger misclassifications beyond the table zones can be considered erroneous. The effect of increased cluster sizes can be seen in Table 3. The random forest achieves throughout the whole experiment a larger number of correct assignments than the artificial neural network. This difference is more noticeable for the smallest cluster size.

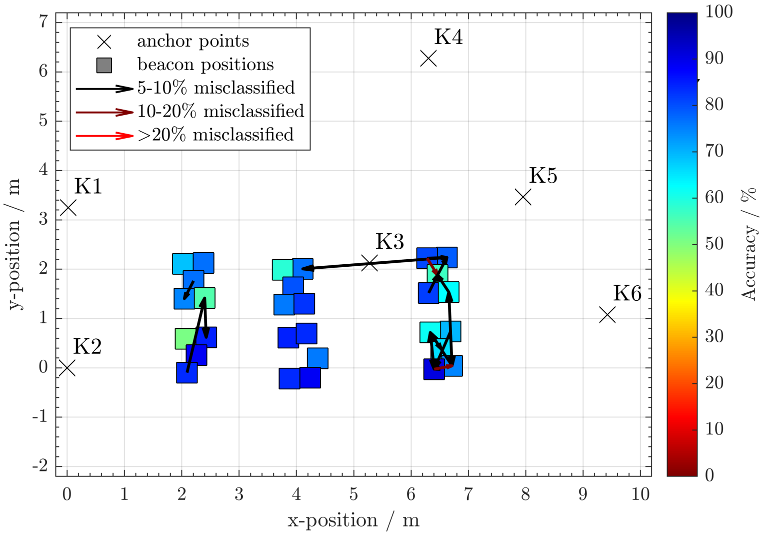


Figure 20. Approximate positions of the clustered beacon positions and the most clear misclassifications using a **neural network**. The colors of the beacon clusters indicate their accuracy of correct assignments.

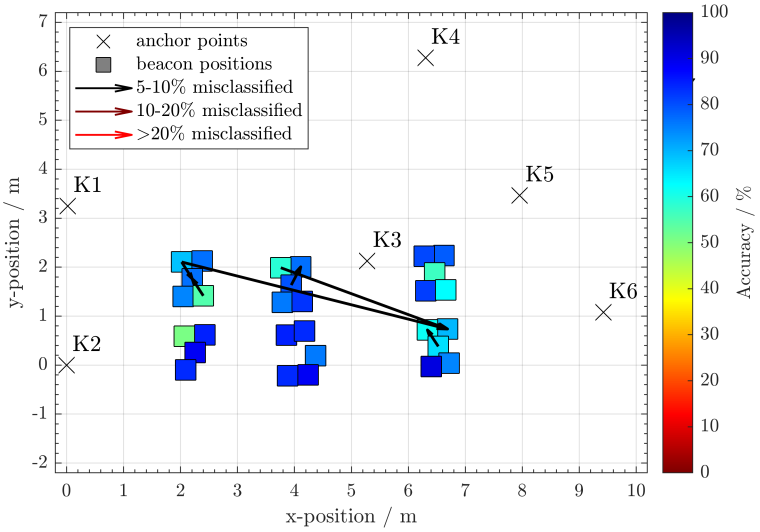


Figure 21. Approximate positions of the clustered beacon positions and the most obvious misclassifications using **random forest**. The colors of the beacon clusters indicate their percentage of correct assignments.

From the results in Table 3 we can conclude that machine learning can be used for localization in situation where one wants to avoid manually measuring the positions of the anchors and requires only to detect whether the speaker is in certain regions.

Table 3. Correct assignments using an artificial neural network and a random forest.

Cluster	Approx. Dimensions	Neural Network	Random Forest
I	0.03 m^2	87.8%	90.5%
II	0.80 m^2	94.1%	94.3%
III	7.15 m^2	95.5%	96.8%

8. Conclusions

In this work, we have presented two methods for acoustic indoor localization which do not require manually measuring the positions of the microphones. We have presented a method for estimating the anchor positions using an inertial measurement unit as additional information. This method provides an initial estimation for non-linear optimization algorithms which often fall into local minima. By doing this, we increase the single-shoot success rate. We have shown the performance of our method in both simulations and real-world experiments. The other approach presented in this work consists of using machine learning to identify in which region a speaker is lying. We have shown experimentally how using random forest classification, one can know where the sender is lying with 94 % of accuracy defining regions of around 0.8 m^2.

However, various open lines for future work remain. When estimating the position of the microphones, we have multiple restrictions on the trajectory for initialization, as one has to start directly beyond one of the anchor nodes and the trajectory must contain straight line segments. In the future, it would be desirable to overcome those restrictions and allow for more general trajectories.

So far, we have assumed to have all the anchors at a known height and performed two-dimensional localization only. An obvious line for future work is, thus, to extend the methods to three dimensions.

Furthermore, the machine learning approach could be extended by training the algorithm with other data such as the amplitude of the received signals or all the reception times of the reflections.

Author Contributions: Conceptualization, F.H.; methodology, J.B. and J.X.; software, J.B., J.X., L.S. and A.B.; validation, D.J.S., F.H., J.B. and I.H.; formal analysis, J.B., D.J.S., L.S., P.K. and J.X.; investigation, J.B., D.J.S., J.X., L.S. and P.K.; resources, J.B., D.J.S., J.X., L.S. and P.K.; data curation, J.B., D.J.S. and P.K.; writing–original draft preparation, J.B.; writing–review and editing, I.H.; visualization, D.J.S.; supervision, L.R. and C.S.; project administration, F.H.; funding acquisition, F.H. and D.J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Fraunhofer Gesellschaft, the state of Baden-Württemberg in the framework of the MERLIN project, as well the German Federal Ministry of Education and Research (BMBF) under grant number FKZ 01IS18011C.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, H.; Darabi, H.; Banerjee, P.; Liu, J. Survey of wireless indoor positioning techniques and systems. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 1067–1080. [CrossRef]
2. Höflinger, F. *Lokalisierungssysteme für die Positionsbestimmung von Personen und Objekten im Innenraum*; Der Andere Verlag: Uelvesbüll, Gemany, 2014.
3. Bahl, P.; Padmanabhan, V.N.; Bahl, V.; Padmanabhan, V. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Tel Aviv, Israel, Israel, 26–30 March 2000; pp. 775–784.
4. Otsason, V.; Varshavsky, A.; LaMarca, A.; De Lara, E. Accurate GSM indoor localization. In Proceedings of the International Conference on Ubiquitous Computing, Tokyo, Japan, 11–14 September 2005; pp. 141–158.
5. Kim, Y.; Shin, H.; Cha, H. Smartphone-based Wi-Fi pedestrian-tracking system tolerating the RSS variance problem. In Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications, Lugano, Switzerland, 19–23 March 2012; pp. 11–19.
6. Simon, N.; Bordoy, J.; Höflinger, F.; Wendeberg, J.; Schink, M.; Tannhäuser, R.; Reindl, L.; Schindelhauer, C. Indoor localization system for emergency responders with ultra low-power radio landmarks. In Proceedings of the 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Pisa, Italy, 11–14 May 2015; pp. 309–314.
7. Ens, A.; Höflinger, F.; Wendeberg, J.; Hoppe, J.; Zhang, R.; Bannoura, A.; Reindl, L.M.; Schindelhauer, C. Acoustic self-calibrating system for indoor smart phone tracking. *Int. J. Navig. Obs.* **2015**, *2015*, 694695. [CrossRef]
8. Höflinger, F.; Zhang, R.; Hoppe, J.; Bannoura, A.; Reindl, L.M.; Wendeberg, J.; Bührer, M.; Schindelhauer, C. Acoustic self-calibrating system for indoor smartphone tracking (assist). In Procedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, NSW, Australia, 13–15 November 2012; pp. 1–9.
9. Yang, L.; Ho, K. An approximately efficient TDOA localization algorithm in closed-form for locating multiple disjoint sources with erroneous sensor positions. *IEEE Trans. Signal Process.* **2009**, *57*, 4598–4615. [CrossRef]
10. Ho, K.; Lu, X.; Kovavisaruch, L.o. Source localization using TDOA and FDOA measurements in the presence of receiver location errors: Analysis and solution. *IEEE Trans. Signal Process.* **2007**, *55*, 684–696. [CrossRef]
11. Gillette, M.D.; Silverman, H.F. A linear closed-form algorithm for source localization from time-differences of arrival. *IEEE Signal Process Lett.* **2008**, *15*, 1–4. [CrossRef]
12. Smith, J.; Abel, J. Closed-form least-squares source location estimation from range-difference measurements. *IEEE Trans. Acoust. Speech Signal Process.* **1987**, *35*, 1661–1669. [CrossRef]
13. Biswas, R.; Thrun, S. A passive approach to sensor network localization. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; Volume 2, pp. 1544–1549.
14. Wendeberg, J.; Hoflinger, F.; Schindelhauer, C.; Reindl, L. Anchor-free TDOA self-localization. In Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation, Guimaraes, Portugal, 21–23 September 2011; pp. 1–10. [CrossRef]

15. Wendeberg, J.; Schindelhauer, C. Polynomial time approximation algorithms for localization based on unknown signals. In Proceedings of the International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, Ljubljana, Slovenia, 13–14 September 2012; pp. 132–143.
16. Wendeberg, J.; Höflinger, F.; Schindelhauer, C.; Reindl, L. Calibration-free TDOA self-localisation. *J. Locat. Based Serv.* **2013**, *7*, 121–144. [CrossRef]
17. Thrun, S. Affine structure from sound. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 4–7 December 2006; pp. 1353–1360.
18. Bordoy, J.; Schindelhauer, C.; Höflinger, F.; Reindl, L.M. Exploiting Acoustic Echoes for Smartphone Localization and Microphone Self-Calibration. *IEEE Trans. Instrum. Meas.* **2019**. [CrossRef]
19. Niitsoo, A.; Edelhäußer, T.; Eberlein, E.; Hadaschik, N.; Mutschler, C. A Deep Learning Approach to Position Estimation from Channel Impulse Responses. *Sensors* **2019**, *19*, 1064. [CrossRef] [PubMed]
20. Feigl, T.; Nowak, T.; Philippsen, M.; Edelhäuβer, T.; Mutschler, C. Recurrent Neural Networks on Drifting Time-of-Flight Measurements. In Proceedings of the 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nantes, France, 24–27 September 2018; pp. 206–212.
21. Zhang, L.; Huang, D.; Wang, X.; Schindelhauer, C.; Wang, Z. Acoustic NLOS identification using acoustic channel characteristics for smartphone indoor localization. *Sensors* **2017**, *17*, 727. [CrossRef] [PubMed]
22. Ebrahimkhanlou, A.; Salamone, S. Single-sensor acoustic emission source localization in plate-like structures: a deep learning approach. In Proceedings of the Health Monitoring of Structural and Biological Systems XII, Denver, CO, USA, 5–8 March 2018; p. 106001O. [CrossRef]
23. Deffenbaugh, M.; Bellingham, J.G.; Schmidt, H. The relationship between spherical and hyperbolic positioning. In Proceedings of the OCEANS 96 MTS/IEEE Conference Proceedings. The Coastal Ocean-Prospects for the 21st Century, Fort Lauderdale, FL, USA, 23–26 September 1996; pp. 590–595.
24. Schott, D.J.; Höflinger, F.; Zhang, R.; Reindl, L.M.; Yang, H. Fuzzy inference system assisted inertial localization system. In Proceedings of the 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Funchal, Portugal, 27–29 June 2017; pp. 89–93. [CrossRef]
25. Raptor-12HS. Available online: https://www.motionanalysis.com/cameras/raptor-12hs/ (accessed on 18 Feb 2020).
26. Bucher, R.; Misra, D. A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System. *VLSI Des.* **2002**, *15*, 507–520. [CrossRef]
27. Cui, X.; Yu, K.; Lu, S. Direction Finding for Transient Acoustic Source Based on Biased TDOA Measurement. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 2442–2453. [CrossRef]
28. Lynch, K.M.; Park, F.C. *Modern Robotics: Mechanics, Planning, and Control*; Cambridge University Press: Cambridge, UK, 2017.