

Article

Fast and Fault-Tolerant Passive Hyperbolic Localization Using Sensor Consensus

Gyula Simon ^{1,*}  and Gergely Zachár ² ¹ Alba Regia Technical Faculty, Óbuda University, 8200 Székesfehérvár, Hungary² Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37212, USA; gergely.zachar@vanderbilt.edu

* Correspondence: simon.gyula@amk.uni-obuda.hu

Abstract: The accuracy of passive hyperbolic localization applications using Time Difference of Arrival (TDOA) measurements can be severely compromised in non-line-of-sight (NLOS) situations. Consensus functions have been successfully used to provide robust and accurate location estimates in such challenging situations. In this paper, a fast branch-and-bound computational method for finding the global maximum of consensus functions is proposed and the global convergence property of the algorithm is mathematically proven. The performance of the method is illustrated by simulation experiments and real measurements.

Keywords: source localization; emitter localization; TDOA; fault tolerant; consensus function

1. Introduction

Determining the location of an object is a key service in a wide range of applications, including navigation [1,2], robotics [3,4], tracking [5,6], safety and security [7,8], health-care [9,10], and defense [11,12], just to name a few. Global Navigation Satellite Systems (GNSSs) can be used in various outdoor applications [1], while for indoor and other GNSS-deprived areas, several other technologies have been proposed, based on, e.g., acoustic signals [8,11–13], radio signals [5,9], visible light [4,14], or images [15,16].

The localization systems utilize a number of spatially distributed anchors, and most often measure quantities directly related to distances, distance differences, angles, or angle differences between the anchors and the target. The location estimate is then calculated using Time of Arrival (TOA), Time Difference of Arrival (TDOA), angle of arrival (AOA), or angle difference of arrival (ADOA) methods, respectively [17].

The TDOA localization, also known as hyperbolic positioning, is a fairly common method in geolocation [1,18,19], sensor networks [8,11,20,21], indoor localization [22,23], or the Internet of Things [24,25]. In the TDOA scenario, differences of distances between the target and the anchors are used, most often calculated from differences of measured times of arrivals of signals (sound or electromagnetic waves) between the anchors and the target, using the propagation speed of the signal. The one-stage or Direct Position Determination (DPD) methods determine the position estimate without explicitly calculating the Time Difference of Arrival (TDOA) values. In contrast, the two-stage methods estimate the TDOA values in stage 1 and then estimate the location from them in stage 2. The location estimation process is also known as position fixing. Although two-stage methods are not optimal, they are asymptotically equivalent to DPD methods and are popular in systems where the transmission of large amounts of raw data between sensors and the central processing unit is prohibitive [26]. This paper focuses on the position fixing of two-stage methods.

The process of position fixing most often involves closed-form solutions [18,27–32], iterative techniques [33,34], or various consensus-based methods [35–40]. These methods yield satisfactory results when the time difference measurements contain only small errors (i.e., measurement noise). However, in practical scenarios, measurements can contain



Citation: Simon, G.; Zachár, G. Fast and Fault-Tolerant Passive Hyperbolic Localization Using Sensor Consensus. *Sensors* **2024**, *24*, 2891. <https://doi.org/10.3390/s24092891>

Academic Editor: Chris Rizos

Received: 20 March 2024

Revised: 23 April 2024

Accepted: 29 April 2024

Published: 30 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

significant errors (outliers), especially when there is no direct line of sight between the anchors and the target, or when the environment produces signal reflections, or when signal detection, and thus time measurements, is not reliable. If such outlier measurements are present, the position estimates of closed-form and iterative techniques will be biased unless additional measures are taken to remove the outliers prior to the estimation [41,42]. Consensus-based methods can provide accurate estimates even when outlier measurements are present, and were successfully utilized in applications where non-line-of-sight conditions and reflections cause unreliable measurements [38,39].

In this paper, a consensus-based method is studied, where the position estimate is determined by identifying the location with the highest value of the consensus function [38]. A fast computation method was recently proposed that was observed to reliably find the global optimum of the consensus function [40]. This paper reviews the consensus function and introduces an updated search method. The contributions of this paper are the following:

- A new, fast evaluation method is introduced.
- The main contribution is the theoretical proof that ensures that the proposed algorithm always finds the global maximum of the consensus function over a finite grid.
- Finally, a comprehensive performance analysis is provided using simulations and real measurements.

The symbols used in this paper are enumerated in the Section: List of Symbols.

2. Related Work

In Section 2.1, the TDOA problem is formulated, with special emphasis on potential large measurement errors, e.g., due to non-line of sight (NLOS). The consensus-function-based solution is reviewed in Section 2.2, and various fast evaluation methods are discussed in Section 2.3.

2.1. The TDOA Localization Problem

The emitter E is located at an unknown position $P_0 = (x_0, y_0, z_0)$ and it emits a signal at an unknown time T_0 . Sensors $S_i, i = 1, 2, \dots, N$ are deployed at known positions $p_i = (x_i, y_i, z_i)$. The emitted signal is detected by sensor S_i at time instant $t_i, i = 1, 2, \dots, N$. If the propagation speed of the signal is c and there is line of sight (LOS) between E and S_i , then the detection time t_i can be expressed as follows:

$$t_i = T_0 + \frac{|P_0 - p_i|}{c}. \quad (1)$$

A measurement scenario is illustrated in Figure 1, where sensors $S_1, S_2, \dots, S_5$ are in LOS of the source, while S_6 and S_7 are in NLOS positions and provide measurements through a reflective surface. In the example, measured times $t_1, t_2, \dots, t_5$ satisfy (1), but for S_6 and S_7 , the measured time is larger than the ideal time calculated from (1).

The objective is to determine the position P_0 using the measurements t_i and positions p_i . The task becomes more challenging when measurements contain outliers, such as those caused by NLOS conditions.

Several solutions have been proposed to solve the TDOA problem. Closed-form solutions are available for a limited number of sensors [27–32], while iterative techniques are used for general cases [33,34]. Particle filters [35,43], Hough transform-based solutions [35–37], and consensus functions [38–40] have also been proposed.

In most TDOA localization problems, the propagation speed of the signal is assumed to be known. In [44], a joint estimation method is proposed, which is capable of estimating both the source location and the propagation speed. The placement of anchors has a significant effect on the localization accuracy. The optimal geometry, considering the communication constraints that are important in practice, has been analyzed in [45].

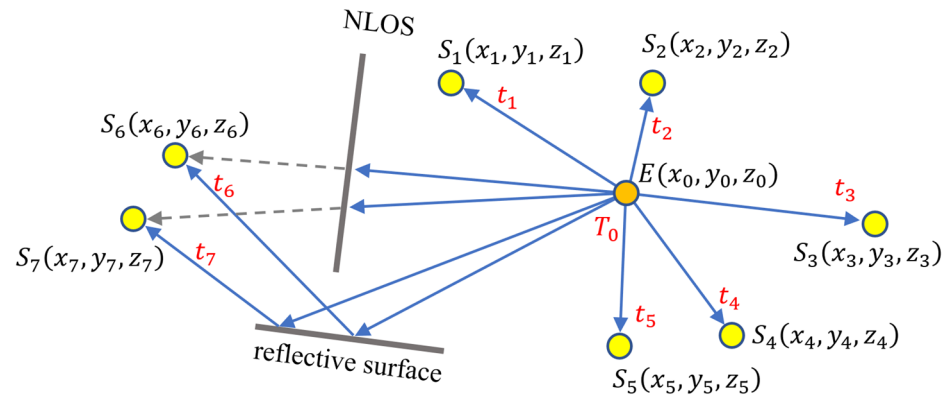


Figure 1. TDOA localization. Emitter E emits an event at unknown time instant T_0 at unknown location (x_0, y_0, z_0) . Sensor S_i measures the time of detection t_i . From the measured t_i and the known sensor positions (x_i, y_i, z_i) , the emitter position is estimated. Sensors S_6 and S_7 measure outliers, due to non-line-of-sight situations.

Most solutions assume that measurements are correct and do not include outliers. In practical cases, however, outliers are common, due to NLOS situations, reflections, sensor faults, etc. Therefore, in real-world scenarios, the presence of outliers can severely bias location estimates unless they are removed from the estimation [41,42].

Several works address the NLOS problem. In [46], an iterative solution, based on the maximum correntropy criterion with a variable center, was proposed. Various methods using convex optimization and semidefinite programming have also been proposed [47–49]. However, these methods are computationally intensive. In [50], a neurodynamic optimization approach is proposed, which provides a trade-off between accuracy and computational complexity.

Other approaches for providing accurate location estimates in the presence of outliers involve sensor consensus. RANSAC-based [51,52] methods randomly select a core group including a small number of sensors (typically 4–5) and use it to perform localization. The measurements from the other sensors are then checked against the core estimate, and the location estimate is refined. In an iterative process, several core groups are selected, and the best result is chosen. By conducting a high number of trials, the results are statistically correct with a high probability [53].

Hough transform-based solutions utilize a voting system in which each pair of sensors votes for a set of possible location estimates. The position with the highest number of votes becomes the location estimate. Fast calculation methods exist when the unknown location is in a plane [37].

Consensus functions also use a voting system, but unlike Hough transform and RANSAC, which use groups of a small number of sensors in the voting process, here, all of the sensors cast a single cooperative vote. The method was successfully utilized in three dimensional acoustic systems with a high number of unreliable measurements [38,39]. The detailed operation is reviewed in Section 2.2.

2.2. Consensus-Function-Based Localization

In this section, consensus functions are reviewed based on [38,39]. The distance d_i between sensor S_i and an arbitrary point $P = (x, y, z)$ is

$$d_i(P) = d_i(x, y, z) = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}. \quad (2)$$

If the source position is (x, y, z) , then the emission time T_0 can be estimated from measurement t_i using (1):

$$T_i(P) = T_i(x, y, z) = t_i - \frac{d_i(x, y, z)}{c}. \quad (3)$$

Thus, $T_i(x, y, z)$ is the estimate of the emission time by a single sensor S_i , provided the source is at location (x, y, z) . Using all sensor measurements, the set of $T_i(x, y, z)$ ($i = 1, 2, \dots, N$) estimates are available for an arbitrary point (x, y, z) . From the set of $T_i(x, y, z)$, the consensus group is formed, according to Definition 1.

Definition 1. A consensus group at position (x, y, z) with window length w contains an S subset of $T_i(x, y, z)$, such that for any two elements $T_j(x, y, z), T_k(x, y, z) \in S$, it is true that $|T_j(x, y, z) - T_k(x, y, z)| < w$.

In the ideal case when there are no measurement errors present, for the true location (x_0, y_0, z_0) , all $T_i(x_0, y_0, z_0)$ estimates are equal to the true emission time T_0 . In real cases when measurement noise is present, the T_i estimates will be in the close vicinity of T_0 , forming a large consensus group. The extent of the ‘vicinity’ is dependent on the level of noise and characterized by the window length w . Sensors that produce outlier measurements may have an estimated value T_i that is significantly different from T_0 , and these estimates will not be members of the consensus group.

An illustration is shown in Figure 2a, where estimates $T_1, T_2, \dots, T_5$ form a five-element consensus group around time T_0 . Estimates T_6 and T_7 , corresponding to outlier measurements, are not part of the consensus group (they form one-element consensus groups).

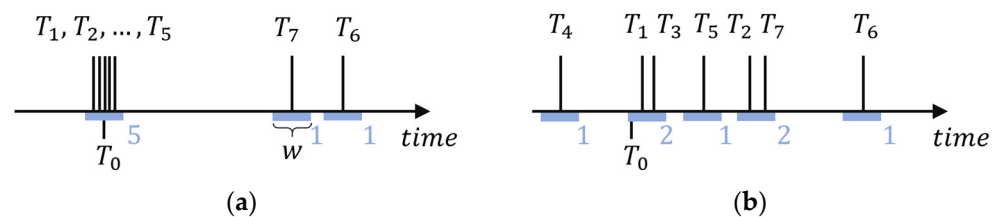


Figure 2. The calculation of the consensus function, where measurements $t_1, t_2, \dots, t_5$ are correct with small measurement noise, while t_6 and t_7 are outliers. (a) Estimated emission times $T_1, T_2, \dots, T_7$ at the true source position. There is a consensus group of five emission time estimates around T_0 ; the value of the consensus function is 5. (b) Estimated emission times at another position. The largest consensus group contains only two estimates; the value of the consensus function is 2. Blue numbers indicate the cardinalities of the consensus groups.

Definition 2. The value of the consensus function at point (x, y, z) is defined as the cardinality of the largest consensus group for location (x, y, z) .

In the example of Figure 2a, the value of the consensus function is 5. For locations (x, y, z) other than (x_0, y_0, z_0) , estimates $T_i(x, y, z)$ are likely to differ significantly, resulting in the formation of small consensus groups rather than one large group. Figure 2b shows an example where the consensus function is evaluated at a point other than the true source location. The estimates T_i are scattered and the largest consensus groups contain only two sensors; thus, the value of the consensus function is 2.

Note that sensors corresponding to a consensus group calculated at point (x, y, z) agree on the hypothesis that the source is located at position (x, y, z) . The more sensors in consensus, i.e., the higher the consensus function, the higher the likelihood that the source is actually at position (x, y, z) . Therefore, the estimated source location is the one with the highest consensus function.

To create consensus groups, the maximum difference w allowed between group members is defined. In Figure 2, a sliding window with width w is shown. The window function Λ_w of width w is defined as follows:

$$\Lambda_w(t) = \begin{cases} 1 & \text{if } |t| < \frac{w}{2} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Using the window function, the consensus function can be expressed as follows:

$$C_w(x, y, z) = \max_{t \in \mathbb{R}} \sum_{i=1}^N \Lambda_w(T_i(x, y, z) - t) \quad (5)$$

The set of points where the consensus function $C_w(x, y, z)$ takes its maximum is

$$\Psi_w = \underset{(x, y, z)}{\operatorname{argmax}}(x, y, z). \quad (6)$$

The location estimate $(\hat{x}_0, \hat{y}_0, \hat{z}_0)$ is calculated as the mean of the positions in Ψ_w :

$$(\hat{x}_0, \hat{y}_0, \hat{z}_0) = \frac{1}{|\Psi_w|} \sum_{p \in \Psi_w} p. \quad (7)$$

The window width w is essential in calculating the consensus function as it determines the required proximity of estimated emission times T_i within a consensus group. At the true location, the estimates T_i should be treated as a single group, despite the small perturbations in the values caused by measurement noise, as shown in Figure 2a. This requirement leads to a lower bound on w , as follows. The accuracy of T_i depends on the accuracy of measurements t_i and the precision of the sensor locations p_i . If the maximum sensor location error is Δs and the maximum time measurement error is $\Delta \tau$, then, from (3), the maximum error of T_i is

$$\Delta T_{\max} = \Delta s/c + \Delta \tau. \quad (8)$$

Since the maximum difference between any two values of T_i is $2\Delta T_{\max}$, this is the lower bound on w . To keep w as small as possible, the window width is set to

$$w = 2\Delta T_{\max} = 2\left(\frac{\Delta s}{c} + \Delta \tau\right). \quad (9)$$

In practical scenarios, a rough a priori estimate of the measurement errors Δs and $\Delta \tau$ is required. From these estimates, the design parameter w can be determined using (9).

2.3. Calculation of the Consensus-Function-Based Location Estimate

The source location is determined by finding the maximum of the consensus function. However, this can be challenging because the consensus function is not smooth, being an integer-valued function, and it may have several local maxima. Traditional gradient-based search methods do not work for such functions. An exhaustive search on a finite grid provides the global maximum; however, for large target areas, especially in 3D, it is not practical. To provide a faster solution, a fast heuristic calculation was proposed in [38,39]. This method is based on integer arithmetic and the Generalized Bisection method and searches for the source location and emission time in a four-dimensional space (x, y, z, t) . The performance of the search method has been experimentally validated [38], but the correctness of the algorithm has not been proven.

Another fast solution was proposed in [53] using a RANSAC-based approach. Random sets of five measurements were utilized to calculate the core estimate, which was then compared to the other measurements using the consensus function. This method is able to provide the correct estimates with high probability if the number of random trials is high enough [53].

A fast Branch and Bound-type calculation method has been proposed recently [40]. A global convergence property of the method has been reported, supported by an experimental validation. The following section presents an improved version of [40] and provides a mathematical proof of its global convergence property.

3. Fast Consensus-Based Localization

The fast search method proposed in [40] performs the search on a grid for the maximum of the consensus function. The grid is iteratively refined until the required grid size is reached. During the refinement process, areas that are unlikely to contain the global maximum are excluded from a further search; thus, the search method converges quickly. The method's correctness was validated through simulation experiments. Now, an enhanced and more detailed version of the algorithm [40] is proposed in Section 3.1. The correctness of the algorithm is proven in Sections 3.2 and 3.3.

3.1. Fast Search on a Finite Grid

The proposed fast search method is described in Algorithm 1, followed by an illustration of its operation.

Algorithm 1 Branch and Bound algorithm to find the maximum of the consensus function

Input:

1. measurements: t_i , $i = 2, 3, \dots, N$
2. sensor positions: (x_i, y_i, z_i) , $i = 1, 2, \dots, N$
3. required (fine) grid size: Δ

Initialization:

4. Define a grid G_f over the search area with a grid cell size equal to Δ . This is the fine grid where the maximum of the consensus function is searched for.
5. Cover the search area with a coarse grid G_c using grid cells of size $2^k \Delta$, where k is an integer greater than 1, so that the grid points of G_c overlap with some of those of G_f . The search starts from this grid.
6. Mark each cell of G_c as active.
7. Calculate the upper bound $\bar{C}_w$ of C_w for each cell

Iteration: repeat Branch and Bound while active cells exist

Branch:

8. Select the active cell with the highest $\bar{C}_w$ value and refer to it as Q . If multiple cells have the same highest value, select the one with the smallest size.
9. Replace Q with smaller cells Q'_i that have a linear size half of the original ($i = 1, 2, 3, 4$ in 2D and $i = 1, 2, \dots, 8$ in 3D).
10. If Q'_i is completely outside of the search area, mark it passive.
11. If the size of Q'_i is larger than Δ then mark Q'_i as active and recalculate the upper bound $\bar{C}_w$ for Q'_i .
12. If the size of Q'_i is equal to Δ then mark Q'_i as final and calculate C_w in the center of the cell

Bound:

13. Calculate the maximum C_{max} of the C_w values in the final cells.
14. Mark active cells with $\bar{C}_w < C_{max}$ as passive

Output:

15. Collect the final cells with $C_w = C_{max}$ to form Ψ_w . Then, calculate $(\hat{x}_0, \hat{y}_0, \hat{z}_0)$.
-

The algorithm finds the maximum of the consensus function using a finite grid with a grid size of Δ (step 3), i.e., in 2D, the grid cells are squares of size $\Delta \times \Delta$ and in 3D, the cells are cubes of size $\Delta \times \Delta \times \Delta$. The consensus function is evaluated in the center of the cells. Note that the grid size determines the resolution of the search.

Figure 3 illustrates the operation, using a 2D example. Figure 3a shows the fine grid over the search space (step 4). Then, a coarse grid (in red) is placed over the fine grid (step 5). The size of the coarse grid must be a 2^k multiple of Δ ; in the example $k = 2$, the coarse cells are thus of size $4\Delta \times 4\Delta$, as shown in Figure 3b. Note that in practical cases, k is typically much larger than the small value used in this example.

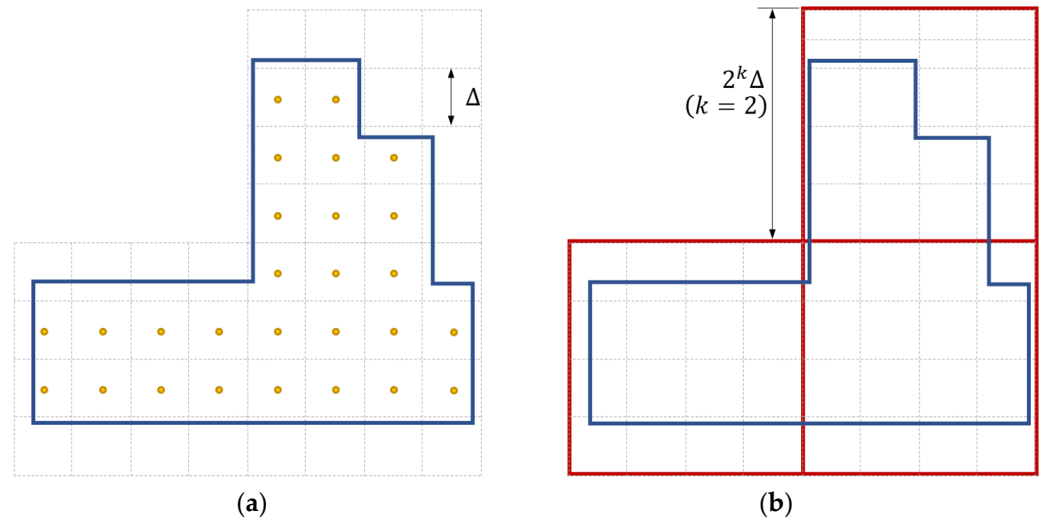


Figure 3. (a) The search area is surrounded by a solid blue line, and the fine search grid is represented by dashed grey lines. The consensus function is to be evaluated at the center points of the search grid, marked by orange dots. (b) The coarse grid, which is represented by the red lines, is placed over the fine grid and used as the initial grid.

The further operation is illustrated in Figure 4. In Figure 4a, the three starting cells are active, indicated by the white background, and the upper bounds $\bar{C}_w$ are displayed within each cell (according to steps 6–7). Figure 4b shows the first branching step. The highest upper bound is seven, so the corresponding cell is selected and divided into four smaller cells (steps 8–9). One cell is outside of the search region and is marked as passive (step 10), shown by an orange background. The new cell size is still larger than Δ , so the upper bounds are calculated for the new cells (step 11). As there are no exact consensus values calculated yet, the bound step is idle. Figure 4c illustrates the subsequent branching step, where the cell with an upper bound of 6 is divided into four smaller cells. The new cell size is now Δ , so the new cells are marked as final (shown by green background) and the exact consensus values are calculated in these cells (step 12). The next bound step is shown in Figure 4d. The value C_{max} is 5 (step 13), and the maximum values are shown by a yellow color in the corresponding cells. Several cells with $\bar{C}_w < 5$ are marked as passive (step 14), shown by an orange background color. Figure 4e shows the next branching step of the active cell with an upper bound of 5. The subsequent bound step is shown in Figure 4f: C_{max} is 5, and thus the cells with upper bound values below 5 are marked as passive. In Figure 4g, the next branch step is shown, where the cell with $\bar{C}_w = 5$ is divided into four final cells. Since there are no more active cells, the bound step is idle and the iteration is finished. Figure 4h shows the cells with $C_w = C_{max} = 5$, forming Ψ_w (step 15).

The design parameter Δ defines the required resolution and thus accuracy, since the search method finds the maximum of the consensus function of the defined grid. Small Δ results in a finer resolution but potentially requires more computation. It is therefore inadvisable to select values for Δ that are excessively small.

In practical situations, the size of Δ can be determined along with parameter w , as follows: The window width w is determined from the time and distance measurement errors, according to (9). The window width w reflects the measurement errors in terms of timing error, while the quantity $c \cdot w$ quantifies the measurement errors in terms of distance. Although the overall positioning accuracy that can be achieved is dependent on the placement of the sensors [44], the order of magnitude can be estimated by $c \cdot w$. Thus, a practically good design choice for grid size Δ lies in the range of $0.1c \cdot w \dots c \cdot w$.

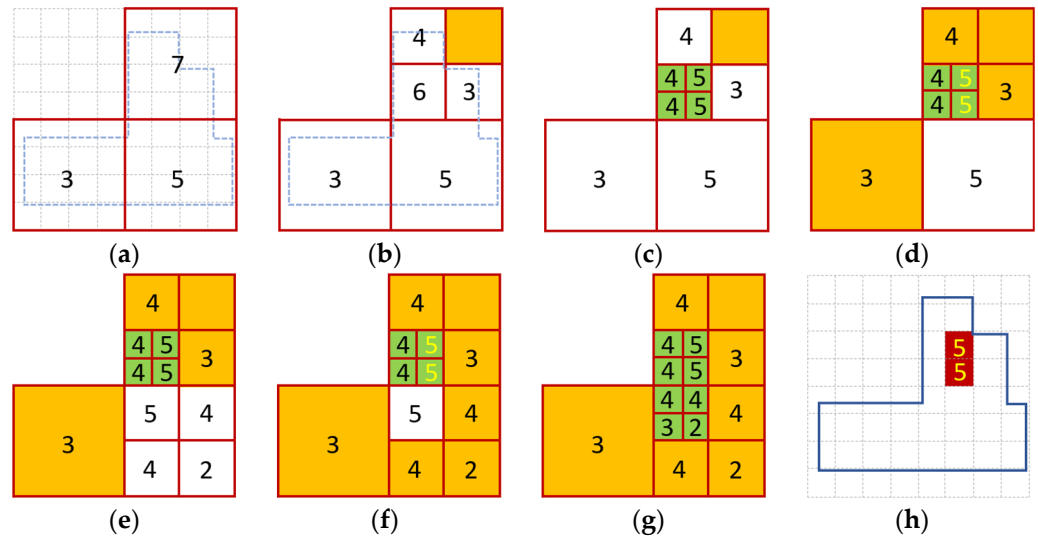


Figure 4. Branch and Bound search to find the maximum of the consensus function. Active, passive, and final cells are denoted by white, orange, and green colors, respectively. Upper bounds are shown in the active and passive cells, and the actual consensus function values are shown in the final cells. The highest consensus values in each iteration are shown by yellow numbers. (a) The fine grid (dashed grey lines) and the coarse initial grid (solid red lines), with the initial upper bounds. The search area is also shown by dashed blue lines. (b) The branch step for the cell with an upper bound of 7. The cell outside of the search region is set to passive. The bound step is idle. (c) The branch step for the cell with an upper bound of 6. The final cell size is reached for the new cells. (d) The bound step. The highest consensus value is 5 (shown by yellow numbers); active cells with an upper bound less than 5 are set to passive (yellow rectangles). (e) The branch step for the active cell with an upper bound of 5. (f) The bound step. The highest consensus value is 5; active cells with a smaller upper bound are set to passive. (g) The branch step for the active cell with an upper bound of 5. There are no more active cells. (h) The cells with the highest consensus value, indicated by a red color, comprise the set of Ψ_w .

3.2. Upper Bound of the Consensus Function

In the operation of Algorithm 1, the upper bound $\bar{C}_w$ of the consensus function in a cell is essential. In [40], a quantity was proposed for $\bar{C}_w$ and it was validated through simulations. However, to date, no proof has been found to support that the proposed quantity is indeed an upper bound. In the following, a mathematical proof will be provided.

The following theorem, Theorem 1, provides an upper bound for the consensus function in a spherical region.

Theorem 1. Let point P be the center of a sphere S with radius r and let Q be a point inside S . If $w' = w + \frac{2r}{c}$, then the following inequality holds:

$$\max_Q C_w(Q) \leq C_{w'}(P). \quad (10)$$

Proof of Theorem 1. Points P , Q , and sensor S_i are shown in Figure 5, along with the sphere S around P . The distance between points P and Q is denoted as d_{PQ} . Let us define the quantity Δd_i as follows:

$$\Delta d_i = d_i(P) - d_i(Q). \quad (11)$$

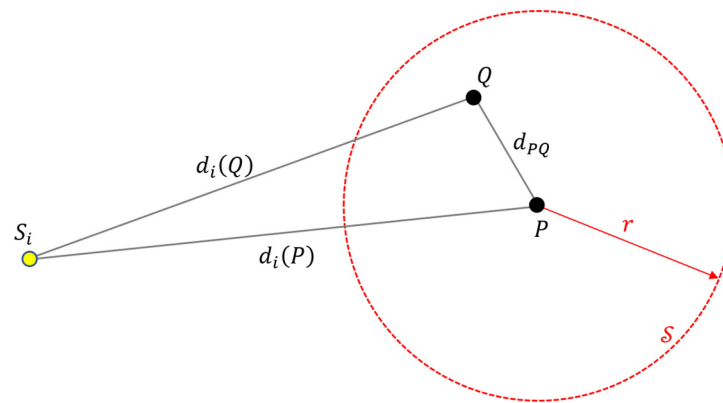


Figure 5. Geometry of points P , Q , and sensor S_i in the proof of Theorem 1.

Since Q is inside S , trivially

$$d_{PQ} \leq r. \quad (12)$$

By applying the triangle inequality to the triangle PQS_i of Figure 5, it also follows that

$$d_{PQ} \geq d_i(P) - d_i(Q). \quad (13)$$

By combining (11)–(13), it follows that

$$\Delta d_i \leq r. \quad (14)$$

Let the value of the consensus function C_w at point Q be denoted by $K = C_w(P)$. Thus, according to (5) and (4), there exists a time instant t_0 and a set Θ of K sensors for which the following holds:

$$|T_i(P) - t_0| < \frac{w}{2}, \quad S_i \in \Theta. \quad (15)$$

For the sake of simplicity, but without loss of generality, let us assume that the sensors in Θ are $S_1, S_2, \dots, S_K$. In the proof, index k refers to these sensors, specifically $k = 1, 2, \dots, K$.

Using (3), the inequality (15) can be written as

$$\left| t_k - \frac{d_k(Q)}{c} - t_0 \right| < \frac{w}{2}. \quad (16)$$

Using (11), the following holds for point P :

$$\left| t_k - \frac{d_k(P)}{c} - t_0 \right| = \left| t_k - \frac{\Delta d_k + d_k(Q)}{c} - t_0 \right| = \left| t_k - \frac{d_k(Q)}{c} - t_0 - \frac{\Delta d_k}{c} \right|. \quad (17)$$

By utilizing the basic inequality $|a - b| < |a| + |b|$, from (17), it follows that

$$\left| t_k - \frac{d_k(P)}{c} - t_0 \right| \leq \left| t_k - \frac{d_k(Q)}{c} - t_0 \right| + \left| \frac{\Delta d_k}{c} \right|. \quad (18)$$

Applying (14) and (16), inequality (18) leads to

$$\left| t_k - \frac{d_k(P)}{c} - t_0 \right| \leq \frac{w}{2} + \frac{r}{c} \quad (19)$$

and thus using (3), it follows that

$$|T_k(P) - t_0| \leq \frac{w}{2} + \frac{r}{c} = \frac{1}{2} \left(w + \frac{2r}{c} \right) = \frac{w'}{2}. \quad (20)$$

Comparing (4) and (20) leads to

$$\Lambda_{w'}(T_k(P) - t_0) = 1. \quad (21)$$

Note that (21) holds for all K sensors $S_1, S_2, \dots, S_K$. Therefore, using (5) and (21), it follows that

$$C_{w'}(P) = \max_{t \in \mathbb{R}} \sum_{i=1}^N \Lambda_{w'}(T_i(P) - t) \geq \sum_{k=1}^K \Lambda_{w'}(T_k(P) - t_0) = \sum_{k=1}^K 1 = K = C_w(Q). \quad (22)$$

Thus, for any point Q inside $\mathcal{S}$, $C_{w'}(P) \geq C_w(Q)$, which leads directly to (10). $\square$

Using the results of Theorem 1, Theorem 2 provides an upper bound for the consensus function in a cell.

Theorem 2. Let us denote the dimension of the search space with D . Let point P_C be the center of a cell $\mathcal{C}$ of size L , where $\mathcal{C}$ is a cube for $D = 3$ or a square for $D = 2$. Then, for all points Q inside $\mathcal{C}$,

$$\max_Q C_w(Q) \leq C_{w_L}(P) \quad (23)$$

with

$$w_L = w + \frac{L\sqrt{D}}{c}. \quad (24)$$

Proof of Theorem 2. For a cube of size L , the maximum distance between its center P_C and any point Q inside the cube is $\frac{L\sqrt{3}}{2}$. For a square of size L , the maximum distance between P_C and any point Q inside the square is $\frac{L\sqrt{2}}{2}$. Thus, for a cell of dimension D ($D = 2, 3$), the cell is inside of a sphere with center P_C and radius $r = \frac{L\sqrt{D}}{2}$. With the choice of

$$w' = w + \frac{2r}{c} = w + \frac{L\sqrt{D}}{c} = w_L, \quad (25)$$

Theorem 1 yields (23). $\square$

3.3. Global Convergence

Finally, the global convergence property of Algorithm 1 is proven.

Theorem 3. If the upper bound $\bar{C}_w$ for a cell with center P and size L is computed as $\bar{C}_w = C_{w_L}(P)$ with $w_L = w + \frac{L\sqrt{D}}{c}$, then Algorithm 1 finds the global maximum of C_w on the finite grid G_f .

Proof of Theorem 3. Let C_{MAX} be the maximum of the consensus function inside the search area, calculated over G_f , and let $\mathcal{C}^*$ denote a particular cell of G_f inside the search area, where the consensus function is C_{MAX} . In Algorithm 1, the initial cells are iteratively divided into smaller cells in the branching steps until the required size of Δ is reached, the cells marked as final being a subset of the cells of G_f . It will be shown that at the end of Algorithm 1, $\mathcal{C}^*$ is in the set of final cells.

In Algorithm 1, the consensus values are calculated in the final cells (step 12), which are the cells of the fine grid G_f . Therefore, for the value C_{max} calculated in step 13, it is true that $C_{max} \leq C_{MAX}$. Note also that for any cell of size L that contains $\mathcal{C}^*$, according to Theorem 2, $\bar{C}_w \geq C_{MAX}$. Therefore, in the bounding phase (steps 13–14) for a cell containing $\mathcal{C}^*$, $\bar{C}_w \geq C_{max}$, so this cell cannot be marked passive in step (14). The same cell cannot be made passive in step 10 either, because $\mathcal{C}^*$ is located inside the search area.

At the beginning of the algorithm, $\mathcal{C}^*$ is part of a larger active cell (step 6). Active cells are eventually (a) divided into smaller active cells (steps 8–9) or (b) smaller final cells

(step 12) or (c) marked as passive (steps 10 and 14). In case (a), C^* is preserved in one of the active cells, while in case (b), C^* becomes one of the final cells. Case (c) does not apply to a cell containing C^* , as was shown above. Thus, a cell containing C^* is either active or final. Since the algorithm always divides the active cells into smaller cells, at the end of the iteration, the cell containing C^* necessarily becomes one of the final cells.

It has been shown that every cell of G_f that gives the maximum consensus function is part of the set of final cells. Since the maximum is searched in this set (step 15), Algorithm 1 finds all of the global maxima over the grid G_f . $\square$

4. Performance Evaluation

The performance of the proposed consensus-function-based method (CF) is evaluated by simulation examples and real measurements. For comparison, we use the iterative LS algorithm and the three-dimensional extension of the COM-W algorithm [32], which provides a closed-form solution.

4.1. Simulations

The simulations utilize a sensor setup that was previously used in a real shooter localization experiment [38]. The bird's eye views of the simulation setups are shown in Figure 6, using 15, 25, and 35 sensors, denoted by grey circles. The elevations of the sensors were in the range of -0.4 m to 10.9 m. The six simulated target positions are shown by red crosses and are listed in Table 1. As Table 1 shows, there were 2D and 3D experiments. In the 2D experiment, the target elevation was known to be at 0 m. This situation is common when the target, such as an autonomous vehicle, is moving on a plane. In the 3D experiments, the target elevation was also to be determined, which is a typical scenario when trying to locate a flying object. It is important to note that the sensor positions were three-dimensional in both the 2D and 3D experiments.

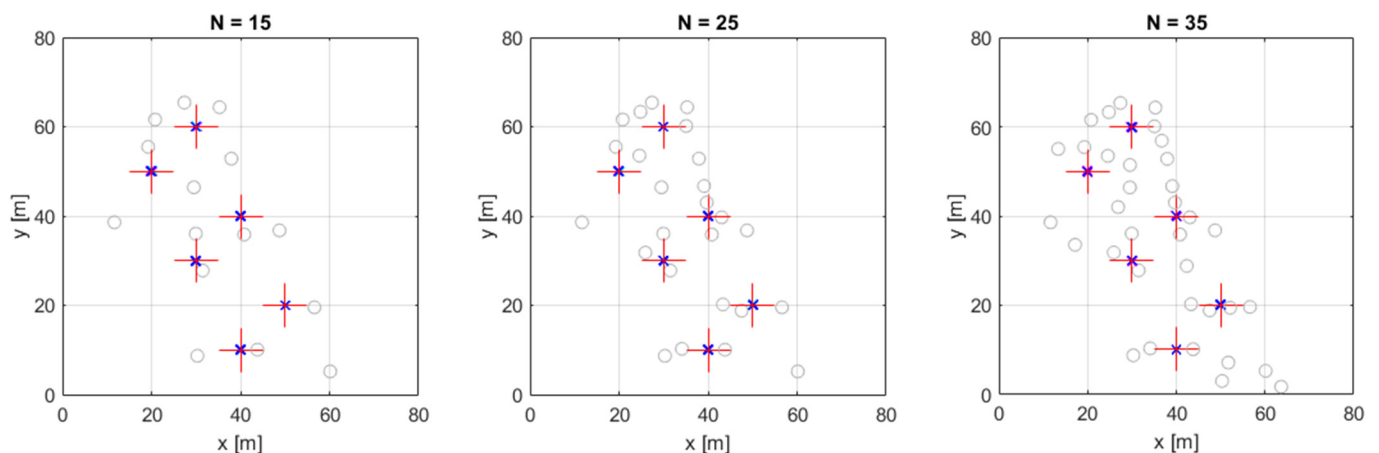


Figure 6. The 3D simulation experiment with additive measurement noise and no outliers. Sensor positions are denoted by grey circles, and red crosses indicate the target positions. Overlapping magenta, cyan, and blue x's show the estimated target positions of LS, COM-W, and CF, respectively.

The simulations were carried out as follows:

- For each target position, we calculated the exact distances d_i between the target position and the sensors.
- We calculated the exact times of arrivals t_i as d_i/c .
- We added measurement noise with normal distribution $\mathcal{N}(0, \sigma)$ to t_i and added additional measurement noise with normal distribution $\mathcal{N}(0, 100\sigma)$ to t_i to emulate the faulty (outlier) sensors. The number of outliers was N_{out} . For each target position, 100 independent measurements were created.

- Using measurements t_i and the sensor positions p_i , the estimated target positions were calculated by LS, COM-W, and CF.
- The tests were conducted using Matlab version R2021b on a computer with i5-8265 CPU with clock frequency of 1.6 GHz, and 24 GB of RAM.
- The LS algorithm was started from a random position within 1 m of the true position.
- The CF method was implemented in Matlab according to Algorithm 1. Apart from Matlab's built-in vector operations, no acceleration methods (e.g., multithreading) were used.
- The final resolution of the CF method was $\Delta = 0.1$ m.

Table 1. Target positions in the simulation experiment.

ID	x (m)	y (m)	z_{2D} (m)	z_{3D} (m)
1	20	50	0	1.5
2	30	30	0	3.1
3	40	10	0	0.5
4	30	60	0	0.2
5	40	40	0	1.1
6	50	20	0	0.1

In the first experiment, there were no outliers present and σ was 0.3 ms (equivalent to a distant measurement error of 0.1 m). The estimated positions for the 3D case are shown in Figure 6 with markers 'x', close to the true positions. Tables 2 and 3 summarize the results for the 2D and 3D experiment, respectively: for each target position, the square root of the Cramer–Rao lower bound (CRLB) is shown, followed by the root-mean-square error (RMSE) of the tested algorithms. In this test, all three methods performed very well with an RMSE close to the theoretical optimum ($\sqrt{CRLB}$). According to the results, there is no practical difference in accuracy among the three methods in this outlier-free experiment.

Table 2. Results of the 2D simulation experiment with measurement noise of $\sigma = 0.1$ m and no outlier measurements.

ID	$N = 15$				$N = 25$				$N = 35$			
	$\sqrt{CRLB}$	RMSE (m)			$\sqrt{CRLB}$	RMSE (m)			$\sqrt{CRLB}$	RMSE (m)		
		LS	COMW	CF		LS	COMW	CF		LS	COMW	CF
1	0.071	0.103	0.086	0.077	0.061	0.090	0.076	0.068	0.048	0.084	0.063	0.060
2	0.068	0.092	0.091	0.077	0.051	0.073	0.057	0.057	0.044	0.079	0.052	0.059
3	0.082	0.092	0.099	0.090	0.067	0.076	0.089	0.072	0.054	0.078	0.067	0.063
4	0.065	0.114	0.098	0.067	0.053	0.083	0.077	0.060	0.047	0.082	0.064	0.059
5	0.064	0.083	0.080	0.065	0.048	0.086	0.061	0.064	0.042	0.078	0.054	0.062
6	0.068	0.072	0.071	0.073	0.057	0.071	0.088	0.064	0.046	0.074	0.057	0.059
mean	0.070	0.092	0.088	0.075	0.056	0.080	0.075	0.064	0.047	0.079	0.060	0.060

In the second experiment, we created outlier measurements as well. We selected the number N_{out} of outliers randomly among $1 \leq N_{out} \leq 6$, independently in each experiment. In each experiment, we generated $N - N_{out}$ correct measurements with a noise level of σ and N_{out} outliers with a noise level of 100σ . The results for the 3D case are shown in Figure 7. Comparing Figures 6 and 7, it is apparent that the LS and COM-W methods have significantly increased variance, with errors occasionally reaching tens of meters. The CF method, however, has a small estimation error. The detailed results for the 2D and 3D cases are listed in Tables 4 and 5, respectively. Clearly, neither the LS nor the COM-W tolerated well the presence of outliers; both algorithms had several meters of errors (the LS solver also diverged several times, but these cases are omitted from the analysis). In 2D, the mean error of LS and COMW methods increased to 2–3 m, while in 3D, the error of these methods

was between 3 and 8 m. The CF method, however, provided low error levels, in 2D, around 0.2 m and in 3D, between 0.3 m and 0.5 m, which was close to the theoretical optimum.

Table 3. Results of the 3D simulation experiment with measurement noise of $\sigma = 0.1$ m and no outlier measurements.

ID	N = 15				N = 25				N = 35			
	$\sqrt{CRLB}$	RMSE (m)			$\sqrt{CRLB}$	RMSE (m)			$\sqrt{CRLB}$	RMSE (m)		
		LS	COMW	CF		LS	COMW	CF		LS	COMW	CF
1	0.27	0.32	0.30	0.27	0.20	0.25	0.21	0.21	0.18	0.26	0.22	0.22
2	0.13	0.15	0.14	0.14	0.11	0.13	0.11	0.11	0.10	0.11	0.11	0.10
3	0.21	0.24	0.33	0.25	0.14	0.15	0.19	0.14	0.12	0.15	0.16	0.12
4	0.30	0.43	0.30	0.45	0.23	0.45	0.23	0.41	0.21	0.46	0.27	0.39
5	0.16	0.19	0.17	0.16	0.10	0.14	0.12	0.12	0.09	0.13	0.12	0.11
6	0.20	0.21	0.22	0.32	0.12	0.14	0.15	0.17	0.10	0.13	0.13	0.15
mean	0.21	0.26	0.24	0.27	0.15	0.21	0.17	0.19	0.13	0.21	0.17	0.18

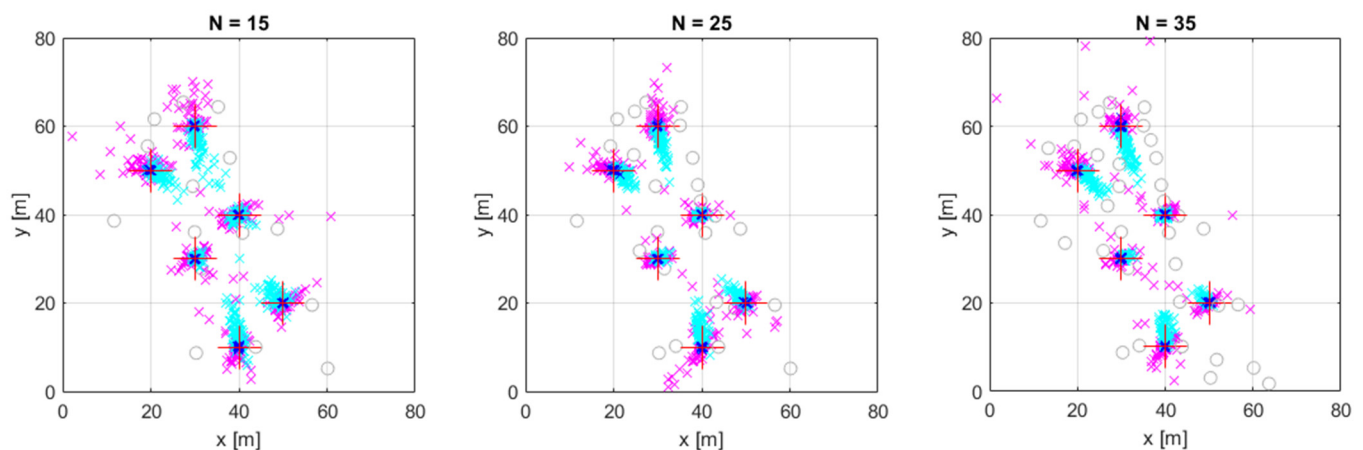


Figure 7. The 3D simulation experiment with additive measurement noise and outliers. Sensor positions are denoted by grey circles, and red crosses indicate the target positions. Magenta, cyan, and blue x's show the estimated target positions of LS, COM-W, and CF, respectively.

Table 4. Results of the 2D simulation experiment with measurement noise of $\sigma = 0.1$ m and $N_{out} = 1 \dots 6$ outlier measurements.

ID	N = 15, $1 \leq N_{out} \leq 3$				N = 25, $1 \leq N_{out} \leq 4$				N = 35, $1 \leq N_{out} \leq 6$			
	$\sqrt{CRLB}$	RMSE (m)			$\sqrt{CRLB}$	RMSE (m)			$\sqrt{CRLB}$	RMSE (m)		
		LS	COMW	CF		LS	COMW	CF		LS	COMW	CF
1	0.071	3.5	3.5	0.13	0.061	2.2	3.6	0.20	0.048	1.7	3.4	0.20
2	0.068	3.4	1.9	0.14	0.051	1.9	1.2	0.12	0.044	3.2	1.1	0.13
3	0.082	3.5	3.8	0.16	0.067	2.8	4.8	0.22	0.054	4.4	3.1	0.15
4	0.065	2.2	3.8	0.22	0.053	1.6	4.0	0.19	0.047	1.9	4.8	0.37
5	0.064	4.4	1.5	0.13	0.048	2.3	0.8	0.12	0.042	2.3	0.7	0.14
6	0.068	2.7	2.9	0.38	0.057	2.9	2.1	0.47	0.046	1.9	1.5	0.15
mean	0.070	3.3	2.9	0.19	0.056	2.3	2.8	0.22	0.046	2.6	2.4	0.19

Table 5. Results of the 3D simulation experiment with measurement noise of $\sigma = 0.1$ m and $N_{out} = 1 \dots 6$ outlier measurements.

ID	$N = 15, 1 \leq N_{out} \leq 3$				$N = 25, 1 \leq N_{out} \leq 4$				$N = 35, 1 \leq N_{out} \leq 6$			
	$\sqrt{CRLB}$	RMSE (m)			$\sqrt{CRLB}$	RMSE (m)			$\sqrt{CRLB}$	RMSE (m)		
		LS	COMW	CF		LS	COMW	CF		LS	COMW	CF
1	0.27	16	9.1	0.40	0.20	4.8	2.7	0.37	0.18	5.2	2.6	0.27
2	0.13	6.6	3.3	0.29	0.11	2.9	1.2	0.27	0.10	18.0	2.3	0.22
3	0.21	7.2	6.5	0.38	0.14	3.9	4.5	0.24	0.12	2.2	3.8	0.25
4	0.30	5.9	7.7	0.88	0.23	6.5	3.5	0.45	0.21	10.1	5.7	0.81
5	0.16	5.7	3.7	0.19	0.10	5.2	2.3	0.44	0.09	4.3	0.95	0.17
6	0.20	9.8	5.4	0.84	0.12	2.6	3.1	0.25	0.10	4.7	1.7	0.37
mean	0.21	8.5	6.0	0.50	0.15	4.3	2.9	0.34	0.13	7.4	2.8	0.35

The distribution of the error is somewhat visible in Figure 7, but for a more comprehensive visualization, Figure 8a presents the cumulative distribution function (cdf) of the localization error for the LS, COM-W, and CF methods. To create the figure, we conducted 1000 simulation experiments for position #1 in 3D, with $N = 25$, $1 \leq N_{out} \leq 4$. The vertical dashed lines show the mean error values, corresponding to the values shown in Table 5.

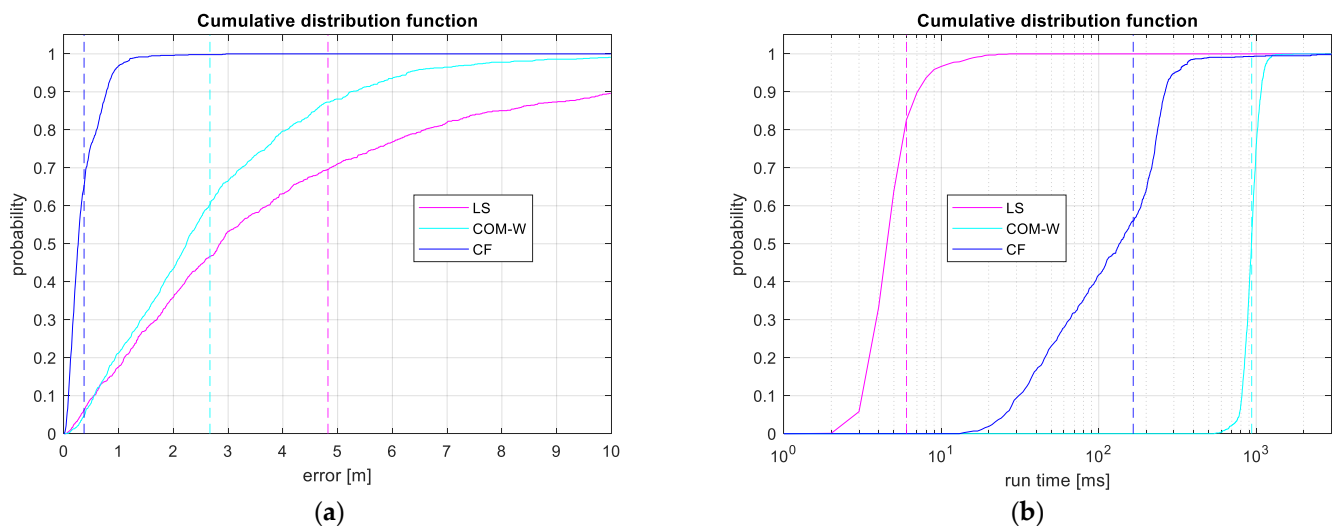


Figure 8. The cumulative distribution function of the localization error in 3D (a) and the execution time (b) at position #1, using 25 sensors and the maximum 4 outliers. The dashed lines show the mean error values and mean execution times.

The comparison of the results of the CF method in the first and second experiments shows that the level of RMSE increases when outliers are present. Note that the theoretical limit (CRLB) corresponds to the case where every sensor provides good measurements, which is not the case here. The reduction in the number of functional sensors led to an increase in the error level.

We measured the mean execution times of the algorithms as a function of the number of sensors and they are presented in Table 6. In the experiments, the LS method was the fastest, with an execution time of approximately 4 ms in the 2D case and 5 ms in the 3D case. The CF method required approximately 20 ms in 2D and 90 ms in 3D. The slowest method was COM-W with an average execution time of more than 200 ms in 2D and 1.3 s in 3D. It is also apparent that the execution time of the COM-W method strongly depends on N , whereas the LS and CF methods did not show significant dependence on it. As an example, the cumulative distribution functions of the execution times for the 3D experiment at position #1 are shown in Figure 8b. The vertical dashed lines show the mean execution

times for this particular experiment. The sharp transition of COM-W indicates that the execution time of COM-W is quite deterministic, with an average of approximately 950 ms. The iterative LS has a run-time that varies between 3 and 20 ms, with a mean value of 6 ms. The execution time of CF is also dependent on the specific scenario, with the majority of values falling between 10 and 400 ms, with a mean value of 165 ms.

Table 6. Execution times of the LS, COM-W, and CF methods in the simulation experiments.

N	Mean Execution Time, 2D (ms)			Mean Execution Time, 3D (ms)		
	LS	COM-W	CF	LS	COM-W	CF
15	3.9	34.8	18.8	5.0	95.6	89.0
25	3.7	164.8	19.8	4.7	814.7	85.8
35	3.6	439.5	20.8	4.7	3225.7	83.1
mean	3.7	213.1	19.8	4.8	1378.7	86.0

The experiment in Figure 9 illustrates the fault tolerance of CF. In the test, we used target position #5 with 35 sensors, and varied the number of outliers from 0 to 25, with the outlier sensors chosen randomly. For each outlier number, we conducted 100 independent experiments, and measured the RMSE for the methods CF, LS, and COM-W. The figure shows the mean estimation errors with solid lines, and the upper and lower edges of the shaded areas correspond to the 90th percentile and the 50th percentile (median) of the error, respectively.

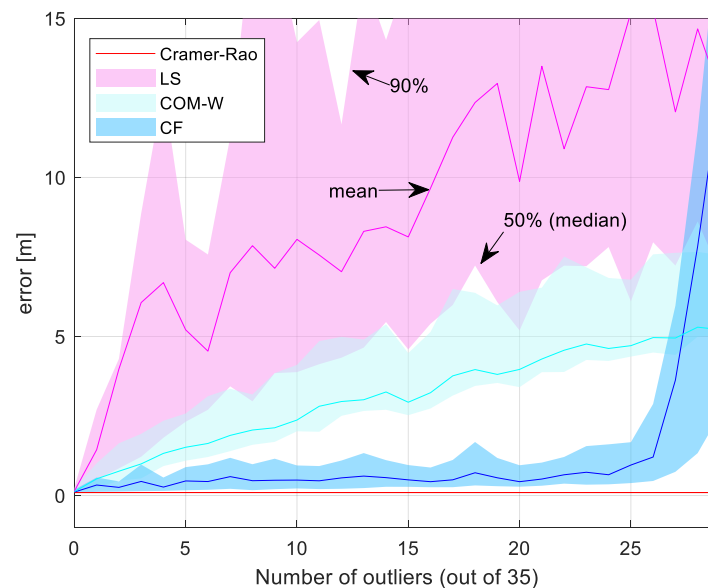


Figure 9. Fault tolerance test results. Solid lines show the mean error, and the shaded regions indicate the error between the 50th and 90th percentiles.

As Figure 9 shows, the error levels for all three methods are close to the Cramer–Rao lower bound, when there are no outliers present. When the first outliers appear, the error of LS and COM-W increases to several meters and the error continues to increase for a higher number of outliers. It is noteworthy that the COM-W method tolerates outliers better than the LS method. The error of the CF method increased slightly with the presence of outliers but remained close to the theoretical value until the number of bad measurements exceeded 25 (i.e., 71% of the total number of 35). Above this outlier number, the performance of the CF method decreased significantly. The ability of the CF method to tolerate more than 70% of sensors being outliers clearly demonstrates its robustness.

4.2. Measurements

The performance of the CF method was evaluated using two real-world measurements. The first measurement was obtained from the public database UTIL [54], where measurements were performed using ultrawideband (UWB) radios. The data were collected in an indoor flight arena of size $7\text{ m} \times 8\text{ m} \times 3.5\text{ m}$. The setup comprised 8 fixed UWB anchors, and the target was a UWB unit mounted on a flying quadrotor platform. The reference trajectory of the target was measured by a millimeter-accuracy optical system, using 10 Vicon Vantage+ cameras. We used measurement record `const4-trial6-tdoa2-traj3`, which contained obstacles (wooden and metal boxes) to create challenging NLOS measurements [54]. The measurement setup is shown in Figure 10, where the grey circles indicate the beacons and the red line represents the target trajectory.

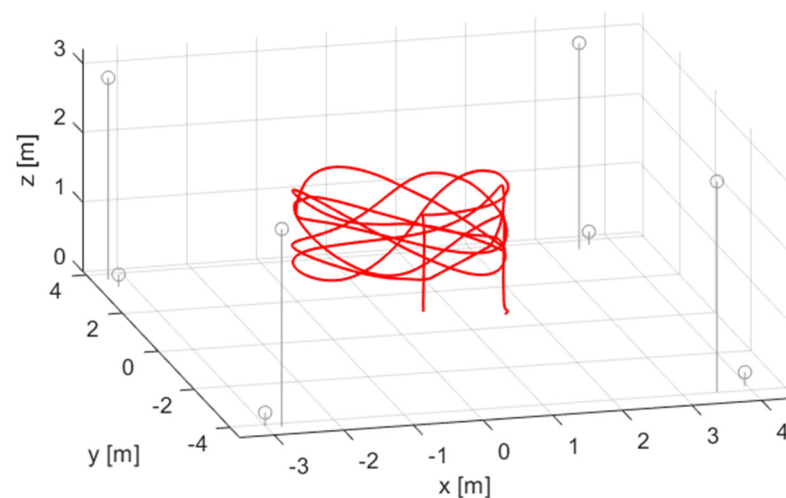


Figure 10. Sensor placement and trajectory of the indoor experiment. Grey circles indicate the sensor positions, and the red curve shows the trajectory of the flying target.

Note that in the tests, we estimated the target position in every measurement position separately, and no filters (e.g., Kalman filters) were utilized to enhance the estimation quality. The estimated target positions are shown as colored dots in Figure 11, for the LS, COM-W, and CF methods. According to the results, the vertical inaccuracy is significantly higher than the inaccuracy in either the x or y direction. This is due to the particular setup of the beacons. The horizontal-only (RMSE-xy) and three-dimensional (RMSE-xyz) values are shown in Table 7. The 2D errors are around 0.2 m, while the 3D error is around 0.5 m. In this experiment, the LS method had the highest RMSE of 0.62 m, while the COM-W handled the NLOS situations better with an RMSE of 0.47 m. The CF provided the most accurate estimates with an error of 0.41 m. Note that the accuracy of the reference algorithm in [54] is 0.45 m, which was also outperformed by CF.

Table 7. Estimation errors and execution times of the LS, COM-W, and CF methods in the indoor localization experiment.

	RMSE-xy (m)	RMSE-xyz (m)	Execution Times (ms)
LS	0.24	0.62	4.2
COM-W	0.19	0.47	4.3
CF	0.15	0.41	11.1

The execution times are also shown in Table 7. In this setup, both LS and COM-W required approximately 4 ms to provide an estimate, while the execution time of CF was around 11 ms.

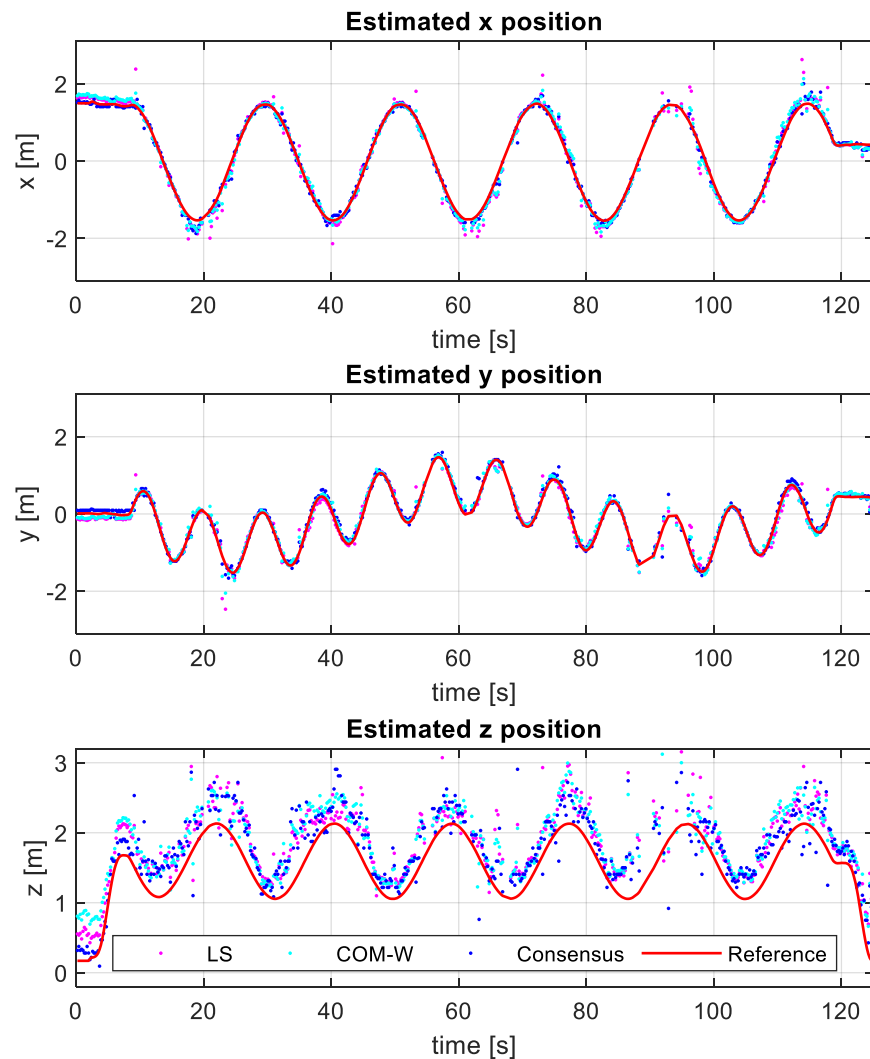


Figure 11. Estimated target positions of the indoor experiment.

The data of the second measurement are from a counter-sniper experiment [38]. The experiment was conducted in a village that had been utilized for military training. The village comprised several streets and numerous buildings. The 57 acoustic sensors were deployed on the street level and the window sills. The shooter fired a sniper weapon from various reference positions and the muzzle blast of the weapon was detected by the sensors. The ten reference shooter positions are listed in Table 8. The sensors are represented by grey circles in the bird's eye view of Figure 12, where the elevations were between -0.4 m and 10.9 m. Note that the setup is a challenging NLOS situation: in each experiment, multiple sensors lacked a direct line of sight. These sensors were either unable to detect the muzzle blast, or they detected signals that were reflected by one or more of the walls of the buildings. The reflected signals arrived significantly later than the signals that should have arrived in a line-of-sight scenario (see Figure 1), so these measurements must be considered outliers.

In Figure 12, the estimated positions are shown by magenta, cyan, and blue x markers for the LS, COM-W, and CF, respectively. The enlarged image illustrates that the CF method produced estimations that closely match the true positions, whereas the LS and COM-W methods had significantly larger errors. The corresponding estimation errors are listed in Table 9, along with the number N of sensors providing measurement results and the consensus function value C_w . In every case, there is a difference between N and C_w , which means measurements with large errors are present. The significant errors are due to the lack of line of sight for a number of the sensors. The outliers cause high estimation errors

for LS and COM-W. The mean error was 6.6 m for COM-W and 6.8 m for LS. The maximum error was 12.5 m for COM-W and 14.2 m for LS. The CF method tolerated outliers much better: the mean error was below 0.9 m, while the maximum error was 1.9 m.

Table 8. Source positions in the counter-sniper experiment.

ID	x (m)	y (m)	z (m)
1	36.34	67.55	3.55
2	30.30	66.42	−0.30
3,4	31.94	57.34	−0.30
5,6,7,8	28.93	45.45	7.30
9	25.85	40.44	−0.20
10	33.37	48.24	−0.25

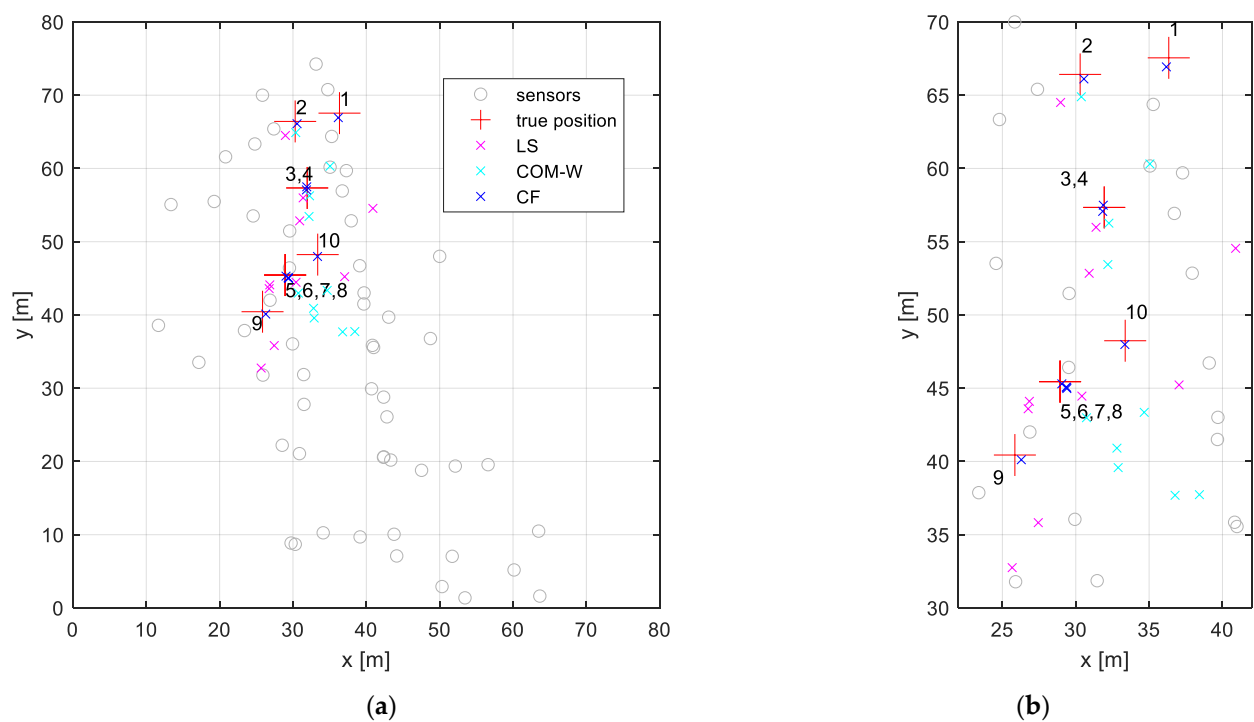


Figure 12. Bird's eye view of the shooter localization experiment. The sensor elevations were in the range of -0.4 m to 10.9 m. (a) Full field and (b) enlarged view around the target positions.

Table 9. Shooter position estimation errors along with the number N of sensors providing measurements and the consensus function value C_w .

ID	Position Estimation Error (m)			Run-Time (ms)			N	C_w
	LS	COM-W	CF	LS	COM-W	CF		
1	14.22	10.04	0.80	105	1404	109	29	25
2	2.34	2.74	1.28	24	11	350	10	9
3	1.50	2.00	1.18	6	69	52	26	23
4	4.65	3.91	0.89	6	2261	92	34	25
5	12.41	8.79	0.34	8	2789	60	38	23
6	3.29	6.13	0.14	6	292	141	21	19
7	13.00	12.51	0.50	4	215	490	20	16
8	3.54	3.37	0.77	5	67	125	15	14
9	7.82	11.37	1.06	6	1159	34	30	21
10	4.82	5.32	1.94	6	1224	79	30	22
average	6.76	6.62	0.89	18	949	153		

The execution times of the algorithms are also shown in Table 9. The LS method was the fastest, with a mean run-time of 18 ms. The CF required 153 ms on average, while for COM-W, the mean execution time was 949 ms. Note that the execution time of COM-W depends on the number N of sensors, while for LS and CF methods, the speed of convergence depends on the actual shape of the error surface, which is hard to predict.

In the case of the indoor measurement, the CF method demonstrated comparable speed with the LS and COM-W methods. In the context of the shooter problem, however, the CF method was significantly slower than the LS method, yet proved to be considerably faster than the COM-W method. The indoor measurement is a typical small-scale problem, whereas the shooter application can be considered a large-scale problem. According to experiments, the CF scales with the number of sensors reasonably well, and the method can be applied for large-scale problems.

The results of the run-time measurements indicate that the execution time of the LS method is only slightly affected by the actual measurement setup, and that it is not directly dependent on the number of sensors. The execution time of the COM-W is influenced by the number of sensors only. The run-time of the CF is in part dependent on the number of sensors and the actual shape of the consensus function. Consequently, the speed of the CF method is the least predictable of the three methods, especially in dynamically changing environments.

The shooter test database allows for a comparison of the proposed CF and previous consensus-based methods. In [38], the Generalized Bisection method was applied to accelerate the computation, while in [53], a mixed RANSAC–consensus approach was utilized. The brute-force computation of the consensus function in the $80\text{ m} \times 80\text{ m} \times 5\text{ m}$ search space with a grid size of $\Delta = 0.1\text{ m}$ would require a 3.2×10^7 evaluation of the consensus function. The Generalized Bisection method was reported to use 10^5 consensus function calls [38], while the RANSAC–consensus solution required 1.8×10^4 function calls to solve the shooter localization problem. The proposed CF method required an average of 5.9×10^3 consensus function calls in the shooter test database, being the most efficient from among the consensus-based approaches. Note that the CF method guarantees that the global maximum is found, while the global convergence of other methods is not guaranteed.

5. Conclusions

A fast and reliable calculation method has been proposed to find the maximum of the consensus function on a finite grid. The algorithm has been theoretically proven to find the global maximum of the consensus function on the grid. The proposed CF method is capable of solving the hyperbolic localization problem even in the presence of a large number of outlier measurements (e.g., from NLOS measurements). The method is extremely robust: it provides estimates with an error level close to the theoretical optimum (Cramer–Rao lower bound) when a large percentage (even as much as 70%) of the measurements are outliers.

The application of the CF method is beneficial in situations where measurements include outliers (e.g., NLOS measurements or bad sensor detections), even a large number of them, and the necessary level of accuracy is still near the theoretical minimum. One disadvantage of the CF method is that its computational cost is higher than that of the simple LS method. Also, its speed depends on the actual error surface; thus, it cannot be predicted and guaranteed. Therefore, the application of the CF method is suggested in soft real-time systems.

The consensus-based approach is not limited to TDOA measurements only. The derivation of hybrid consensus functions including TDOA and angle of arrival (AOA) or angle difference of arrival (ADOA) measurements requires further research. Another topic for future work is the more efficient implementation of the proposed method through the use of parallel implementation.

Author Contributions: Conceptualization, G.S.; methodology, G.S. and G.Z.; validation, G.S. and G.Z.; formal analysis, G.S.; writing, G.S. and G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The UTIL Ultra-wideband Dataset used in this study is available at the following public domain database: <https://utiasdsl.github.io/util-uwbd-dataset/> (accessed on 7 February 2024).

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

List of Symbols

List and definition of symbols.

Notations	Definition
c	propagation speed
C_w	consensus function with window w
$\bar{C}_w$	upper bound of C_w
D	dimension of the search space
$d_i(P) = d_i(x, y, z)$	distance between P and S_i
G_c	coarse grid
G_f	fine grid
$P_0 = (x_0, y_0, z_0)$	source position
$p_i = (x_i, y_i, z_i)$	position of sensor $\#i$
S_i	sensor $\#i$
T_0	emission time
$T_i(P) = T_i(x, y, z)$	emission time estimated from t_i , provided the source is at P
t_i	detection time at sensor $\#i$
$d_i(P) = d_i(x, y, z)$	distance between P and S_i
w	window length
$(\hat{x}_0, \hat{y}_0, \hat{z}_0)$	location estimate
Δ	grid size of G_f
Δs	maximum sensor location error
$\Delta \tau$	maximum time measurement error
Λ_w	window function
Ψ_w	set of points where the consensus function takes its maximum

References

1. Lechner, W.; Baumann, S. Global navigation satellite systems. *Comput. Electron. Agric.* **2000**, *25*, 67–85. [\[CrossRef\]](#)
2. Indelman, V.; Gurfil, P.; Rivlin, E.; Rotstein, H. Real-time vision-aided localization and navigation based on three-view geometry. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2239–2259. [\[CrossRef\]](#)
3. Placed, J.A.; Strader, J.; Carrillo, H.; Atanasov, N.; Indelman, V.; Carlone, L.; Castellanos, J.A. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Trans. Robot.* **2023**, *39*, 1686–1705. [\[CrossRef\]](#)
4. Guan, W.; Chen, S.; Wen, S.; Tan, Z.; Song, H.; Hou, W. High-Accuracy Robot Indoor Localization Scheme Based on Robot Operating System Using Visible Light Positioning. *IEEE Photonics J.* **2020**, *12*, 7901716. [\[CrossRef\]](#)
5. Lin, H.; Zhan, J. GNSS-denied UAV indoor navigation with UWB incorporated visual inertial odometry. *Measurement* **2023**, *206*, 112256. [\[CrossRef\]](#)
6. Zhou, Z.; Feng, W.; Li, P.; Liu, Z.; Xu, X.; Yao, Y. A fusion method of pedestrian dead reckoning and pseudo indoor plan based on conditional random field. *Measurement* **2023**, *207*, 112417. [\[CrossRef\]](#)
7. Gao, S.; Liu, J.; Zong, Y.; Wang, M.; Jin, X.; Tian, G.; Dai, X. Blast source TDOA localization with time synchronization estimation based on spatial overpressure-monitoring network. *Measurement* **2022**, *204*, 112080. [\[CrossRef\]](#)
8. Sun, S.; Zhao, C.; Zheng, C.; Zhao, C.; Wang, Y. High-Precision Underwater Acoustical Localization of the Black Box Based on an Improved TDOA Algorithm. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 1317–1321. [\[CrossRef\]](#)
9. Mercuri, M.; Sacco, G.; Hornung, R.; Zhang, P.; Visser, H.J.; Hijdra, M.; Liu, Y.H.; Pisa, S.; van Liempd, B.; Torfs, T. 2-D localization, angular separation and vital signs monitoring using a SISO FMCW radar for smart long-term health monitoring environments. *IEEE Internet Things J.* **2021**, *8*, 11065–11077. [\[CrossRef\]](#)

10. Bibbò, L.; Carotenuto, R.; Della Corte, F. An Overview of Indoor Localization System for Human Activity Recognition (HAR) in Healthcare. *Sensors* **2022**, *22*, 8119. [\[CrossRef\]](#)
11. Maroti, M.; Simon, G.; Ledeczki, A.; Sztipanovits, J. Shooter localization in urban terrain. *Computer* **2004**, *37*, 60–61. [\[CrossRef\]](#)
12. Abiri, A.; Parsayan, A. The Bullet Shockwave-Based Real-Time Sniper Sound Source Localization. *IEEE Sens. J.* **2020**, *20*, 7253–7264. [\[CrossRef\]](#)
13. Kundu, T. Acoustic source localization. *Ultrasonics* **2014**, *54*, 25–38. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Bai, L.; Yang, Y.; Chen, M.; Feng, C.; Guo, C.; Saad, W.; Cui, S. Computer Vision-Based Localization with Visible Light Communications. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 2051–2065. [\[CrossRef\]](#)
15. Wolf, J.; Burgard, W.; Burkhardt, H. Robust vision-based localization by combining an image-retrieval system with Monte Carlo localization. *IEEE Trans. Robot.* **2005**, *21*, 208–216. [\[CrossRef\]](#)
16. Simon, G.; Zachár, G.; Vakulya, G. Lookup: Robust and Accurate Indoor Localization Using Visible Light Communication. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2337–2348. [\[CrossRef\]](#)
17. Widdison, E.; Long, D.G. A Review of Linear Multilateration Techniques and Applications. *IEEE Access* **2024**, *12*, 26251–26266. [\[CrossRef\]](#)
18. Ho, K.C.; Chan, Y.T. Solution and performance analysis of geolocation by TDOA. *IEEE Trans. Aerosp. Electron. Syst.* **1992**, *29*, 1311–1322. [\[CrossRef\]](#)
19. Li, J.; Lv, S.; Jin, Y.; Wang, C.; Liu, Y.; Liao, S. Geolocation and Tracking by TDOA Measurements Based on Space–Air–Ground Integrated Network. *Remote Sens.* **2023**, *15*, 44. [\[CrossRef\]](#)
20. Wang, Y.; Ho, K.C. TDOA positioning irrespective of source range. *IEEE Trans. Signal Process.* **2017**, *65*, 1447–1460. [\[CrossRef\]](#)
21. Sun, Y.; Zhang, F.; Wan, Q. Wireless Sensor Network-Based Localization Method Using TDOA Measurements in MPR. *IEEE Sens. J.* **2019**, *19*, 3741–3750. [\[CrossRef\]](#)
22. Do, T.H.; Yoo, M. TDOA-based indoor positioning using visible light. *Photon Netw. Commun.* **2014**, *27*, 80–88. [\[CrossRef\]](#)
23. Wang, M.; Chen, Z.; Zhou, Z.; Fu, J.; Qiu, H. Analysis of the Applicability of Dilution of Precision in the Base Station Configuration Optimization of Ultrawideband Indoor TDOA Positioning System. *IEEE Access* **2020**, *8*, 225076–225087. [\[CrossRef\]](#)
24. Zhao, K.; Zhao, T.; Zheng, Z.; Yu, C.; Ma, D.; Rabie, K.; Kharel, R. Optimization of Time Synchronization and Algorithms with TDOA Based Indoor Positioning Technique for Internet of Things. *Sensors* **2020**, *20*, 6513. [\[CrossRef\]](#)
25. Wang, G.; Zhu, W.; Ansari, N. Robust TDOA-Based Localization for IoT via Joint Source Position and NLOS Error Estimation. *IEEE Internet Things J.* **2019**, *6*, 8529–8541. [\[CrossRef\]](#)
26. Swindlehurst, A.L.; Stoica, P. Maximum likelihood methods in radar array signal processing. *Proc. IEEE* **1998**, *86*, 421–441. [\[CrossRef\]](#)
27. 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. [\[CrossRef\]](#)
28. Chan, Y.T.; Ho, K.C. A simple and efficient estimator for hyperbolic location. *IEEE Trans. Signal Process.* **1994**, *42*, 1905–1915. [\[CrossRef\]](#)
29. Mahajan, A.; Walworth, M. 3D position sensing using the differences in the time-of-flights from a wave source to various receivers. *IEEE Trans. Robot. Autom.* **2001**, *17*, 91–94. [\[CrossRef\]](#)
30. 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. [\[CrossRef\]](#)
31. Liu, N.; Xu, Z.; Sadler, B.M. Low-Complexity Hyperbolic Source Localization with a Linear Sensor Array. *IEEE Signal Process. Lett.* **2008**, *15*, 865–868. [\[CrossRef\]](#)
32. Cao, S.; Chen, X.; Zhang, X.; Chen, X. Combined Weighted Method for TDOA-Based Localization. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1962–1971. [\[CrossRef\]](#)
33. Foy, W.H. Position-Location Solutions by Taylor-Series Estimation. *IEEE Trans. Aerosp. Electron. Syst.* **1976**, *AES-12*, 187–194. [\[CrossRef\]](#)
34. Zhou, Z.; Rui, Y.; Cai, X.; Lu, J. Constrained total least squares method using TDOA measurements for jointly estimating acoustic emission source and wave velocity. *Measurement* **2021**, *182*, 109758. [\[CrossRef\]](#)
35. Mikhalev, A.; Ormondroyd, R.F. Comparison of Hough Transform and Particle Filter Methods of Emitter Geolocation using Fusion of TDOA Data. In Proceedings of the 2007 4th Workshop on Positioning, Navigation and Communication, Hannover, Germany, 22 March 2007; pp. 121–127. [\[CrossRef\]](#)
36. Mikhalev, A.; Hughes, E.J.; Ormondroyd, R.F. Comparison of Hough Transform and particle filter methods of passive emitter geolocation using fusion of TDOA and AOA data. In Proceedings of the 2010 13th International Conference on Information Fusion, Edinburgh, UK, 26–29 July 2010; pp. 1–8. [\[CrossRef\]](#)
37. Simon, G.; Leitold, F. Passive TDOA Emitter Localization Using Fast Hyperbolic Hough Transform. *Appl. Sci.* **2023**, *13*, 13301. [\[CrossRef\]](#)
38. Simon, G.; Maróti, M.; Lédeczi, A.; Balogh, G.; Kusy, B.; Nádas, A.; Pap, G.; Sallai, J.; Frampton, K. Sensor network-based countersniper system. In Proceedings of the SenSys '04 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; pp. 1–12. [\[CrossRef\]](#)
39. Lédeczi, Á.; Nádas, A.; Völgyesi, P.; Balogh, G.; Kusy, B.; Sallai, J.; Pap, G.; Dóra, S.; Molnár, K.; Maróti, M.; et al. Countersniper system for urban warfare. *ACM Trans. Sens. Netw.* **2005**, *1*, 153–177. [\[CrossRef\]](#)

40. Simon, G.; Vakulya, G. Fast Calculation Method for Time Difference of Arrival-based Localization. In Proceedings of the 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nuremberg, Germany, 25–28 September 2023; pp. 1–5. [\[CrossRef\]](#)
41. Compagnoni, M.; Pini, A.; Canclini, A.; Bestagini, P.; Antonacci, F.; Tubaro, S.; Sarti, A. A Geometrical–Statistical Approach to Outlier Removal for TDOA Measurements. *IEEE Trans. Signal Process.* **2017**, *65*, 3960–3975. [\[CrossRef\]](#)
42. Apolinário, J.A.; Yazdanpanah, H.; Nascimento, A.S.; de Campos, M.L.R. A Data-selective LS Solution to TDOA-based Source Localization. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 4400–4404. [\[CrossRef\]](#)
43. Khalaf-Allah, M. Particle Filtering for Three-Dimensional TDoA-Based Positioning Using Four Anchor Nodes. *Sensors* **2020**, *20*, 4516. [\[CrossRef\]](#)
44. Zou, Y.; Liu, H. TDOA Localization With Unknown Signal Propagation Speed and Sensor Position Errors. *IEEE Commun. Lett.* **2020**, *24*, 1024–1027. [\[CrossRef\]](#)
45. Sadeghi, M.; Behnia, F.; Amiri, R. Optimal Geometry Analysis for TDOA-Based Localization Under Communication Constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3096–3106. [\[CrossRef\]](#)
46. Wang, W.; Wang, G.; Ho, K.C.; Huang, L. Robust TDOA localization based on maximum correntropy criterion with variable center. *Signal Process.* **2023**, *205*, 108860. [\[CrossRef\]](#)
47. Wang, W.; Wang, G.; Zhang, F.; Li, Y. Second-order cone relaxation for TDOA-based localization under mixed LOS/NLOS conditions. *IEEE Signal Process. Lett.* **2016**, *23*, 1872–1876. [\[CrossRef\]](#)
48. Su, Z.; Shao, G.; Liu, H. Semidefinite programming for NLOS error mitigation in TDOA localization. *IEEE Commun. Lett.* **2018**, *22*, 1430–1433. [\[CrossRef\]](#)
49. Ma, X.; Ballal, T.; Chen, H.; Aldayel, O.; Al-Naffouri, T.Y. A Maximum-Likelihood TDOA Localization Algorithm Using Difference-of-Convex Programming. *IEEE Signal Process. Lett.* **2021**, *28*, 309–313. [\[CrossRef\]](#)
50. Xiong, W.; Schindelhauer, C.; So, H.C.; Bordoy, J.; Gabbrielli, A.; Liang, J. TDOA-based localization with NLOS mitigation via robust model transformation and neurodynamic optimization. *Signal Process.* **2021**, *178*, 107774. [\[CrossRef\]](#)
51. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [\[CrossRef\]](#)
52. Zhuang, Y.; Sun, X.; Li, Y.; Huai, J.; Hua, L.; Yang, X.; Cao, X.; Zhang, P.; Cao, Y.; Qi, L.; et al. Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches. *Inf. Fusion* **2023**, *95*, 62–90. [\[CrossRef\]](#)
53. Vakulya, G.; Simon, G. Fast Adaptive Acoustic Localization for Sensor Networks. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 1820–1829. [\[CrossRef\]](#)
54. Zhao, W.; Goudar, A.; Qiao, X.; Schoellig, A.P. UTIL: An ultra-wideband time-difference-of-arrival indoor localization dataset. *arXiv* **2022**, arXiv:2203.14471. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.