

## Article

# A Geometric Approach to Noisy EDM Resolution in FTM Measurements <sup>†</sup>

Jerome Henry <sup>1,\*</sup> , Nicolas Montavont <sup>2</sup>, Yann Busnel <sup>2</sup> , Romaric Ludinard <sup>2</sup>  and Ivan Hrasko <sup>3</sup>

<sup>1</sup> Cisco Systems, Research Triangle Park, NC 27560, USA

<sup>2</sup> IMT Atlantique, IRISA, 35510 Cesson-Sévigné, France; nicolas.montavont@imt-atlantique.fr (N.M.); yann.busnel@imt-atlantique.fr (Y.B.); romaric.ludinard@imt-atlantique.fr (R.L.)

<sup>3</sup> Cisco Systems, 821 09 Bratislava, Slovakia; ihrasko@cisco.com

\* Correspondence: jerhenry@cisco.com; Tel.: +1-919-508-7894

<sup>†</sup> This paper is an extended version of our paper published in 16th Wireless and Mobile Computing, Networking And Communications (WiMob 2020).

**Abstract:** Metric Multidimensional Scaling is commonly used to solve multi-sensor location problems in 2D or 3D spaces. In this paper, we show that such technique provides poor results in the case of indoor location problems based on 802.11 Fine Timing Measurements, because the number of anchors is small and the ranging error asymmetrically distributed. We then propose a two-step iterative approach based on geometric resolution of angle inaccuracies. The first step reduces the effect of poor ranging exchanges. The second step reconstructs the anchor positions, starting from the distances of highest likely-accuracy. We show that this geometric approach provides better location accuracy results than other Euclidean Distance Metric techniques based on Least Square Error logic. We also show that the proposed technique, with the input of one or more known location, can allow a set of fixed sensors to auto-determine their position on a floor plan.

**Keywords:** FTM; fine timing measurement; 802.11; mMDS; euclidean distance metric; multidimensional scaling



**Citation:** Henry J.; Montavont N.; Busnel Y.; Ludinard R.; Hrasko I. A Geometric Approach to Noisy EDM Resolution in FTM Measurements. *Computers* **2021**, *10*, 33. <https://doi.org/10.3390/computers10030033>

Academic Editors: Anna Maria Vegni, Marc Kurz and Chunxiao Jiang

Received: 31 January 2021

Accepted: 9 March 2021

Published: 12 March 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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/>).

## 1. Introduction

With Global Positioning System (GPS) augmented with fusion techniques (cellular trilateration, dead reckoning, etc.), outdoor location has become a reality on a large part of the planet. Location is more challenged indoor. The ability for a personal device (phone, tablet) to display the user location is useful, but GPS availability is often limited indoor.

### 1.1. FTM Approach to Indoor Location

Alternative techniques have been sought for decades. Among them, 802.11-2016 [1] Fine Timing Measurement (FTM), expanded in the 802.11az amendment, defines a procedure for indoor location tracking based on Wi-Fi ranging. Wi-Fi is massively deployed all over the planet. With common spectrum available worldwide, more than 16 billion devices in use and more than 4 billion new supporting devices shipped each year, according to the Wi-Fi Alliance, it is expected that any personal device (phone or tablet) would support Wi-Fi, and that Wi-Fi would be available in about any public place, anywhere. As such, a location method relying on Wi-Fi technologies has all the chances of seeing a large adoption, and is therefore of high interest. With FTM, an initiating station (ISTA) exchanges frames with a responding station (RSTA) and uses the times of the exchanges to deduce its own position relative to the RSTA. Specifically, after negotiating the exchange parameters, ranging measurements operate through bursts. At the beginning of each burst, the RSTA sends a frame (at time  $t_1$ ) that reaches the ISTA at time  $t_2$ . The ISTA sends an acknowledgement frame at time  $t_3$ , that reaches the RSTA at time  $t_4$ . In the subsequent

frame, the RSTA communicates the  $t_1$  and  $t_4$  values. The ISTA can now compute the time of flight of the exchange and its distance  $d$  to the RSTA by a simple:

$$d = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \times c \quad (1)$$

where  $c$  refers to the speed of light. This computation is only valid if a number of conditions are met:

1. The drift in timestamp measurement at the RSTA is linear. The RSTA provides a time value for its frames, but a clock time difference between the ISTA and the RSTA is inconsequential as only  $(t_4 - t_1)$  is considered (thus the interval between the associated frames, not the local time at which each value was measured), removing the need for synchronized clocks. However, a difference in drift rate between the RSTA and ISTA clocks may cause the  $(t_4 - t_1)$  and  $(t_3 - t_2)$  intervals to be counted at different tic speeds [2], leading to time and therefore range imprecision.
2. Each side can measure accurately the time of arrival (ToA) and the time of departure (ToD) of each frame. This time is measured as the arrival (or departure) at the antenna of the beginning of the first training symbol (the beginning of the frame preamble). As the part of the system that measures time is not located in the physical antenna, each system needs to be calibrated to account for the delay between the antenna and the time measurement module. This calibration is usually done in Line of Sight (LoS) conditions. Different factors (temperature, noise, etc.) may render this calibration only partially accurate, thus resulting in larger estimated distances than the ground truth [3].
3. Measurements are done in LoS conditions. The distance between the ISTA and the RSTA should be a straight line. If the considered signal is instead a reflection, then distance accuracy is lost. 802.11az includes mechanisms to detect Line of Sight vs non-Line-of-Sight (nLoS) signals. 802.11-2016 FTM does not include such mechanism, causing in some cases the receiver to associate the ToA with the first strong reflected component instead of the weaker primary LoS component. In some other scenarios, the signal traverses a strong obstacle where  $c$  is different than in the air. With a technique where each nanosecond represents a travelled distance of approximately a foot (30 cm), such mismatch can contribute to ranging inaccuracies in a measurable way.

All these elements play a role in ranging accuracy and need to be accounted for, as will be shown in this work.

Once computation of the range to multiple RSTAs has been completed, the ISTA only knows its distance to a set of anchors, but does not know its location. FTM allows the RSTA to send its Location Configuration Information (LCI) to the ISTA, a geo-position and height element which content follows a format defined in RFC 6225 [4].

### 1.2. The Challenges of FTM for Indoor Location and this Article Contribution

The FTM procedure introduces the very problem that it aims to solve. Outdoor, Global Positioning System (GPS) technologies provide location information to mobile devices in most environments where LoS to GPS satellites can be established. Indoor, GPS becomes difficult to use for the same reason and devices cannot reliably establish their geo-position anymore. But FTM supposes that RSTAs, positioned indoor, can be configured with their correct geo-position. A common assumption is that these RSTAs would be Wi-Fi access points (APs). In fact, any other type of 802.11 device could support the RSTA function (digital signage, printers, Wi-Fi cameras, etc.). We will call all these objects sensors for simplicity. In most cases, these sensors would be positioned at a static location (or a location that changes only occasionally). Yet configuring their geo-position is difficult. These objects typically do not include a GPS sensor. Even if they would, GPS is challenged indoor. Many implementers are left with manual and time consuming techniques, where the geo-positions of points at the edge of the building are first established (using mobile

devices GPS functions outside, or high-resolution reference aerial pictures), then high precision ranging techniques (laser, Ultra Wide Band, etc.) are used to finally come to an evaluation of each sensor position and location. The process has to be reiterated each time an RSTA is moved or added.

Yet, as soon as the geo-position of one or more device is known, this paper shows that the other sensors can use FTM to self-locate one another, considerably simplifying the deployment of an FTM-ready infrastructure. Solving this problem implies seeding the system with one or more initial positions, which is trivial and can be achieved with a mobile device ranging to one or more RSTAs from outside the building. The solution also implies solving the issue of self-positioning, where the various anchors can reliably establish their respective position based on noisy environments. Thus, the contributions of this article are as follows: noting that pairwise FTM distance noise is not symmetrical, this paper first proposes to reduce the noise by identifying its contribution to geometric distortion of the triangles formed by AP sets, in a phase called Wall Remover. By identifying the effect of inner walls of FTM exchanges between AP pairs, this phase allows for the reduction of the effect of walls on FTM measurements. In a second phase, a recursive examination of all possible Euclidean distance matrices is conducted, to identify the anchors least affected by noise. By combining standard EDM techniques with geometric projections, this phase allows the selection of the best sub-matrices (least effected by remaining measurement noises) to compute the positions of anchors which are most likely to be accurate. From their position, the location of all other anchors is computed, leading to a better position accuracy than EDM resolution techniques alone.

An early version of this work was presented in [5]. In this article, we deepen our proof and extend the applicability of the method to different types of buildings. The rest of this article is organized as follows: Section 2 briefly examines possible techniques to augment GPS for indoor location. Section 4 details the general problem space of noisy Euclidian matrix completion. Section 5 explains our approach, with an asymmetry reduction phase and an algebro-geometric approach. Section 6 presents numerical validation through experimental measurements, and compares the accuracy of our proposed method to other classical Euclidean Distance Matrix (EDM) resolution methods. Section 7 concludes this paper.

## 2. Related Work

Outdoor location based on GPS can commonly provide accuracy down to a meter or less [3]. When nearing a building, the accuracy gets diluted as the building hides some GPS satellites from view [6]. Different augmentation techniques were proposed to limit this issue. For example, Liu et al. [7] proposed using detection of the building and comparison with publicly available aerial imagery to revert to meter precision. Other authors propose to use the known degradation pattern of the signal [8,9] to adjust the location calculation. Our measurements in this paper show that accuracy can be maintained at a sub-meter level in many cases, even near buildings. This will prove useful, as the method proposed in this paper uses outdoor GPS measurements as a final seed to reorient the graph formed by APs having computed their relative positions (but not knowing the position of the graph in the building). Indoor however, GPS accuracy depends heavily on the building structure (windows, floor size and plan etc.). In some scenarios, location can be obtained [7], but in most, the signal is lost. Other technologies are then commonly used to deploy anchors with known positions and perform measurements (signal or distance) against them [10,11]. FTM, studied in this paper, is one of such technologies.

Although FTM is intended for indoor location, the technology allows a mobile device, still outside and detecting GPS satellites, to range and seed its location to one or more indoor RSTAs. The idea of seeding GPS measurements from outdoor to indoor systems is not new, and has for example been studied in [12] and [13]. Once they have ranged to an initial known outdoor location, these RSTAs could range to one another and determine their relative position. Such task falls into the general domain of sensor location in a noisy

environment. This field has been explored widely in multiple contexts, as it represents the need for several systems to evaluate their distance to one another, thus building a distance matrix, and then resolving the inconsistencies between measurements. In wireless technologies, the distance can be evaluated with a variety of techniques (time of flight measurements, signal estimations), but most of them are noisy, resulting in an imperfect, or incomplete, distance matrix. Resolving such matrix is a complex problem, which challenges are summarized in [14]. Ref [15] provides an overview of the main resolution techniques. In the particular case of sensor location, [16] shows that the problem can be convex if the dimension space is known, which is often the case for RSTAs in FTM (but not when using FTM alone, as we will show). An approach close to ours, to reduce the problem with sub-matrix spaces, is detailed in [7], although the proposed method is strictly algebraic (while we propose a geometric component). As the distances are organized in a matrix, algebraic approaches are natural. Some authors, like Doherty and al. [17], explore geometric resolutions in scenarios where link directionality matters. We will show that the addition of learning machine can considerably enhance the resulting accuracy.

### 3. Multidimensional Scaling (MDS) Background

This work focuses on FTM exchanges between fixed points. In this context, Wi-Fi access points and other static devices (e.g., digital signage) can be configured to play the role of ISTAs and RSTAs, alternating between one and the other and ranging to one another over time. As the points are fixed, and as each ranging burst takes a few hundreds of milliseconds (e.g., a burst of 30 exchanges would consume about 250 ms in good RF conditions), a large number of samples can be taken (e.g., a burst per minute represents 43,200 samples per 24-hour period). This flexibility allows for obtaining ranges between pairs far apart, and for which only a few exchanges succeed. From all exchanges, only the best (typically smallest) distance is retained, as will be seen later. The output of these FTM measurements is a network of  $n > 0$  nodes, among which  $0 \leq p \leq n$  nodes have a known position.

The measured distance between these nodes can be organized in a matrix that we will note  $\tilde{D}$ . Conceptually, the set is similar to any other noisy Euclidean Distance Matrix (EDM). The set contains an exhaustive table of distances  $\tilde{d}_{ij}, \forall i, j \in 1, \dots, m$  between points taken by pairs from a list of  $0 < m \leq n$  points  $x$  in  $N$  dimensions ( $x_i$  and  $x_j \in \mathbb{R}^N$  for any point  $x_i$ ). Each point is labelled ordinally, hence each row or column of an EDM, i.e.,  $\forall (i, j) \in \llbracket 0; m \rrbracket^2$ , individually addresses all the points in the list. The main task of the experimenter is then to find the dimension  $N$  and construct a matrix of distances that on one hand best resolves the noise (which causes inconsistencies between the various measured pairs), and on the other hand is closest to the real physical distances between points, called ground truth, and which matrix is noted  $D$ . Such task is one main object of Multidimensional Scaling (MDS).

MDS draws its origin from psychometrics and psychophysics. MDS considers a set of  $n$  elements and attempts to evaluate similarities or dissimilarities between them. These properties are measured by organizing the set elements in a multi-dimensional geometric object where the properties under evaluation are represented by distances between the various elements. MDS thus surfaces geometrical proximity between elements displaying similar properties in some dimension of an  $\mathbb{R}^N$  space. The properties can be qualitative (non-metric MDS, nMDS), where proximity may be a similarity ranking value, or quantitative (metric MDS, mMDS), where distances are expressed. This paper will focus on mMDS.

Two main principles lie at the heart of mMDS: the idea that distances can be converted to coordinates, and the idea that during such process dimensions can be evaluated. The coordinate matrix  $X$  is an  $n * N$  object, where each row  $i$  expresses the coordinate of the point  $i$  in  $N$  dimensions. Applying the squared Euclidean distance equation to all points  $i$  and  $j$  in  $X$  (Euclidean distance  $d_{ij}(X)$ , which we write  $d_{ij}$  for simplicity) allows for an interesting observation:

$$d_{ij}^2(X) = d_{ij}^2 = \sum_{a=1}^N (x_{ia} - x_{ja})^2 = \sum_{a=1}^N (x_{ia}^2 + x_{ja}^2 - 2x_{ia}x_{ja}) \quad (2)$$

Applying this equation to the distance matrix  $D$ , noting  $D^{(2)}$  the squared distance matrix for all  $d_{ij}$ ,  $c$  an  $n \times 1$  column vector of all ones,  $x_a$  the column  $a$  of matrix  $X$ , noting  $e$  a vector that has elements  $\sum_{a=1}^N x_{ia}^2$ , and writing  $A'$  the transpose of any matrix  $A$ , the equation becomes:

$$D^{(2)} = ec' + ce' - 2 \sum_{a=1}^N x_a x_a' = ec' + ce' - 2XX'. \quad (3)$$

This observation is interesting, in particular because it can be verified that the diagonal elements of  $XX'$  are  $\sum_{a=1}^N x_{ia}^2$ , i.e., the elements of  $e$ . Thus, from  $X$ , it is quite simple to compute the distance matrix  $D$ . However, mMDS usually starts from the distance matrix  $D$ , and attempts to find the position matrix  $X$ , or its closest expression  $U$ . This is also FTM approach, that starts from distances and attempt to deduce location. As such, mMDS is directly applicable to FTM.

Such reverse process is possible, because one can observe that  $D$ , being the sum of scalar products of consistent dimensions, is square and symmetric (and with only positive values). This observation is intuitively obvious, as  $D$  expresses the distance between all point pairs, and thus each point represents one row and one column in  $D$ . These properties are very useful, because a matrix with such structure can be transformed in multiple ways, in particular through the process of eigendecomposition, by which the square, positive and symmetric matrix  $B$  of size  $n \times n$  can be decomposed as follows:  $B = Q\Lambda Q'$ , where matrix  $Q$  is orthonormal (i.e.,  $Q$  is invertible and we have  $Q^{-1} = Q'$ ) and  $\Lambda$  is a diagonal matrix such that  $\forall (i, j) \in \llbracket 1; n \rrbracket, \lambda_{i,j} = 0 \forall i \neq j$  and  $\lambda_{ii} \neq 0$ . Values  $\lambda_{ii}, \forall i \in \llbracket 1; n \rrbracket$  of  $\Lambda$  are the eigenvalues of  $B$ . Eigenvalues are useful to find eigenvectors, which are sets of individual non-null vectors  $u_i$  that satisfy the property:  $Bu_i = \lambda_i u_i$ , i.e., the direction of  $u_i$  does not change when transformed by  $B$ .

In the context of MDS, this decomposition is in fact an extension of general properties of Hermitian matrices, for which real and symmetric matrices are a special case. A square matrix  $B$  is Hermitian if it is equal to its complex conjugate transpose  $B^*$ , i.e.,  $B = B^*$ .

In terms of the matrix elements, such equality means that  $\forall (i, j) \in \llbracket 1; n \rrbracket, b_{ij} = b_{ji}$ . A symmetric MDS matrix in  $\mathbb{R}^N$  obviously respects this property, and has the associated property that the matrix can be expressed as the product of two matrices, formed from one matrix  $U$  and its transpose  $U'$ , thus  $B = UU'$ . Because this expression can be found,  $B$  is said to be positive semi-definite (definite because it can be classified as positive or negative, positive because the determinant of every principal submatrix is positive, and positive semi-definite if 0 is also a possible solution). A positive semi-definite matrix has non-negative eigenvalues.

This last property is important for mMDS and for the FTM case addressed in this paper. As  $B_{ij} = B_{ji}$  can be rewritten as  $(B - \lambda_i I)u_i = 0$ , it follows that non-null eigenvectors are orthogonal. As such, the number of non-null eigenvalues is equal to the rank of the matrix, i.e., its dimension. This dimension is understood as the dimension of the object studied by MDS. With FTM, this outcome determines if the graph formed by APs ranging once another is in 2 dimensions (e.g., all APs on the same floor, at ceiling level) or in 3 dimension (e.g., APs on different floors). In a real world experiment,  $B$  is derived from  $\tilde{D}$  and is therefore noisy. But one interpretation of the eigendecomposition of  $B$  is that it approximates  $B$  by a matrix of lower rank  $k$ , where  $k$  is the number of non-null eigenvalues, which can then represent the real dimensions of the space where  $B$  was produced. This is because, if  $q_i$  is the  $i$ -th column vector of  $Q$  (and therefore  $q_i'$  the  $i$ -th row vector of  $Q'$ ), then  $B = Q\Lambda Q'$  can be written as:

$$B = [\lambda_1 q_1 \ \lambda_2 q_2 \ \dots \ \lambda_n q_n] \begin{bmatrix} q'_1 \\ q'_2 \\ \vdots \\ q'_n \end{bmatrix} \quad (4)$$

And thus

$$B = \lambda_1 q_1 q'_1 + \lambda_2 q_2 q'_2 + \dots + \lambda_n q_n q'_n \quad (5)$$

Therefore, if  $n - k$  eigenvalues are 0, so is their individual  $\lambda_j q_j q'_j$  product, and an image of  $B$  can be written as:

$$C = \lambda_1 q_1 q'_1 + \lambda_2 q_2 q'_2 + \dots + \lambda_k q_k q'_k \quad (6)$$

$C$  and  $B$  have the same non-null eigenvectors, i.e.,  $C$  is a submatrix of  $B$  restricted to the dimension of  $B$ 's non-null eigenvalues. In noisy matrices, where distances are approximated, it is common that all eigenvalues will be non-null. However, the decomposition should expose large eigenvalues (i.e., values that have a large effect on found eigenvectors) and comparatively small eigenvalues (i.e., values that tend to reduce eigenvectors close to the null vector). Small eigenvalues are therefore ignored and considered to be null values that appear as non-zero because of the matrix noise.

An additional property listed above is that positive semi-definite matrices only have non-negative eigenvalues. In the context of MDS, this is all the more logical, as each eigenvector expresses one dimension of the  $\mathbb{R}^N$  space. However, noisy matrices may practically also surface some negative eigenvalues. The common practice is to ignore them for rank estimation, and consider them as undesirable but unavoidable result of noise. We will apply the same principles for measurements obtained with FTM. However, it is clear that the presence of many and/or large negative eigenvalues is either a sign that the geometric object studied under the MDS process is not Euclidian, or that noise is large, thus limiting the possibilities of using the matrix directly, without further manipulation.

Therefore, if from Equation (3) above, one defines  $B = XX'$ , then an eigendecomposition of  $B$  can be performed as  $B = Q\Lambda Q'$ . As scalar product matrices are symmetric and have non-negative eigenvalues, one can define  $\Lambda^{1/2}$  as a diagonal matrix with diagonal elements  $\lambda_i^{1/2}$ . From this, one can write  $B = (Q\Lambda^{1/2})(Q\Lambda^{1/2})' = UU'$ . The coordinates in  $U$  differ from those in  $X$ , which means that they are expressed relative to different coordinate systems. However, they represent the geometric object studied by MDS with the same dimensions and scale. Thus, a rotation can be found to transform one into the other.

With these transformations, mMDS can convert a distance matrix into a coordinate matrix, while estimating the space dimension in the process. Thus, classical mMDS starts by computing, from all measured distances, the squared distance matrix  $\tilde{D}^{(2)} = [\tilde{d}_{ij}^2]$ . The measured distance matrix  $\tilde{D}$  is also commonly called, in MDS in general, the proximity matrix, or the similarity matrix. Next, a matrix called the *centering matrix*  $J$  is computed, that is in the form  $J = I_n - \frac{1}{n}cc'$  where  $c$  is an  $n \times 1$  column vector of all ones. Such matrix is a set of weights which column or row-wise sum is twice the mean of the number of entries  $n$ . This matrix has useful properties described in [18]. In particular, applied to  $\tilde{D}$ , it allows the determination of the centered matrix  $B = -\frac{1}{2}J\tilde{D}^{(2)}J$ , which is a transformation of  $\tilde{D}$  around the mean positions of  $\tilde{d}_{ij} \in \mathbb{R}^N$ . This can be seen as follows:

$$-\frac{1}{2}J\tilde{D}^{(2)}J = -\frac{1}{2}J(ec' + ce' - 2XX')J \quad (7)$$

$$= -\frac{1}{2}Jec'J - \frac{1}{2}Jce'J + \frac{1}{2}J(2B)J \quad (8)$$

By transposing  $J$  into the expression, it can easily be seen that, as  $e'J = 0$  and  $Je = 0$ :



$$-\frac{1}{2}J\tilde{D}^{(2)}J = B \quad (9)$$

At this point, the distance matrix is centered. This phase has a particular importance for the method proposed in this paper, because we will see below that its direct effect is to dilute the noise of one pair into the measurements reported by other pairs, thus causing mMDS to fail in highly asymmetric measurements like FTM. mMDS then computes the eigendecomposition of  $B = Q\Lambda Q'$ . Next, the experimenter has the possibility to decide of the dimensions of the projection space ( $\mathbb{R}^2$  or  $\mathbb{R}^3$  in our case, but the dimension can be any  $m \in N$  in mMDS). This can be done by arbitrarily choosing the  $m$  largest eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_m$  and their corresponding eigenvectors of  $B$ . This choice is useful when the experimenter decides to project the distance matrix into  $m$  dimensions, and has decided of what value  $m$  should be. Alternatively, the experimenter can observe all eigenvalues in  $B$  and decide that the dimension space  $\mathbb{R}^m$  matches all  $m$  large positive eigenvalues in  $B$ , ignoring the (comparatively) small positive eigenvalues, along with the null and negative eigenvalues as detailed in the previous section.

Then, if we write  $\Lambda_m$  the matrix of these  $m$  largest positive eigenvalues, and  $Q_m$  the first  $m$  columns in  $Q$  (thus the matrix of eigenvalues matching the dimensions decided by the experimenter), the coordinate matrix is determined to be  $U = Q_m\Lambda_m^{1/2}$ .

#### 4. Problem Space Framework

##### 4.1. mMDS Limitations in FTM Measurements

We note that mMDS is a family of techniques, with multiple possible variations, but all of them abide by the general core principles expressed in the previous section. These principles present properties of great value for multiple distance applications, but also three unsolvable limitations for indoor measurements like FTM: pairwise asymmetry is ignored, dimension space cannot be determined, and as the errors are centered, location accuracy is always typically poor. This section examines these issues in turn.

##### 4.1.1. Pairwise Asymmetry

Although FTM does build a square distance matrix  $\tilde{D}$ , the FTM distance matrix displays several specific properties. One such property is asymmetry. In most cases, measurements are unidirectional, one side acting as the initiator, the other as the responder. In the case of static anchors, the roles can be inverted at the conclusion of each exchange. The process repeats for each sensor pair. These exchanges are affected by localities. As the measurements are collected in a noisy environment, they suffer from the side effects of multipath. Although most sensors attempt to determine the first signal to establish the line of sight (and therefore shortest distance), it is common that strong multipath may drown that signal to a point where the receiver identifies the first strong reflection as the first signal. In an environment where the position of reflective surfaces cannot be predicted, such effect may affect differently each receiver. As a consequence, the initial distance matrix set is asymmetric, where, for many pairs,  $\tilde{d}_{ij} \neq \tilde{d}_{ji}$ .

If obstacles or reflection sources are static (e.g., walls), such asymmetry cannot be resolved by continuous measurements or by averaging over a long period (asymmetry persists). In an implementation where FTM sensors are calibrated (i.e., where the computed distance in line of sight is close to, but never less than, the ground truth), a simple corrective measure consists to only consider the smaller value for each reported pair distance, thus allowing  $\tilde{d}_{ij} = \tilde{d}_{ji} = \min_{(i,j)}\{\tilde{d}_{ij}, \tilde{d}_{ji}\}$ .

##### 4.1.2. Dimension Determination

This approximation solves single pair distance inequalities, but indoor measurements surface another type of asymmetry. As illustrated in Figure 1, indoor walls are local objects. As the signal passes through a wall, it can be slowed down enough to induce an increased travel time between the ISTA and the RSTA, and thus a measured distance  $\tilde{d}_{ij}$  dilated by

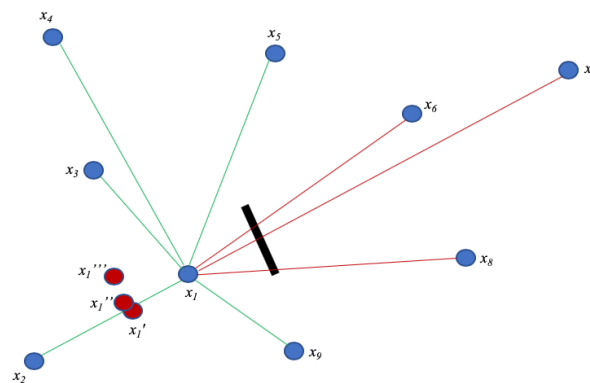
a factor  $k_{ij}$  (compared to the ground truth  $d_{ij}$ ) different than the dilation for another pair, between which a wall would not be found, and thus, for 3 APs  $i, j$  and  $l$ :

$$\frac{\tilde{d}_{ij}}{d_{ij}} \neq \frac{\tilde{d}_{jl}}{d_{jl}} \neq \frac{\tilde{d}_{il}}{d_{il}} \quad (10)$$

Algebraically, a similar dilation factor  $k$  for all measured distances would be ideal and would make the dilation easy to resolve. Each measured distance would then be a simple (dilated) single scalar representation of the ground value matrix. As:

$$\tilde{D} = \begin{bmatrix} \tilde{d}_{11} & \dots & \tilde{d}_{1n} \\ \dots & \dots & \dots \\ \tilde{d}_{n1} & \dots & \tilde{d}_{nn} \end{bmatrix} = \begin{bmatrix} kd_{11} & \dots & kd_{1n} \\ \dots & \dots & \dots \\ kd_{n1} & \dots & kd_{nn} \end{bmatrix} = k \begin{bmatrix} d_{11} & \dots & d_{1n} \\ \dots & \dots & \dots \\ d_{n1} & \dots & d_{nn} \end{bmatrix} = kD \quad (11)$$

Then  $\tilde{D}$  would be a simple homothetic transformation of  $D$ , and our only task would be to find the scalar  $k$ , the dilation factor. But in real measurements, noise is not linear and each  $\tilde{d}_{ij}$  is affected by a different dilation. Finding individual  $k_{ij}$  values becomes challenging, even if we reduce the scope by half by making the hypothesis that  $d_{ij} = d_{ji} = \min\{d_{ij}, d_{ji}\}$  as above (thus considering that the smallest distance is likely closest to the true LoS distance), and thus that  $k_{ij} = k_{ji}$ . This geometry of the indoor space results in two additional major difficulties when attempting to use mMDS.



**Figure 1.** Directionality of measurement errors.

The first difficulty relates to dimension definition. FTM measurements are collected either in 2 dimensions (all APs or sensors at the same level, e.g., at ground or ceiling level on the same floor) or 3 dimensions (multi-floor scenario). One task is therefore to determine the dimensions of the collection space, expecting 2 or 3 as a result. However, the interference locality expressed before results in different inaccuracies between measurements. This problem is difficult to resolve. A common MDS approach in this case is to let the experimenter decide of the dimensions, then use the least square error technique to project the matrix in the assigned  $\mathbb{R}^n$  space. But this approach is only valid if the experimenter knows *a priori* the dimension  $n$ . In an indoor setting where RF travels in all directions and where distances are only measured by time of flight, such an arbitrary decision puts an unacceptable burden of knowledge on the FTM experimenter. Without knowing *a priori*  $k_{ij}$ , the determination of the vertical component of a set of sensors is only possible if  $k_{ij}$  happens to be small relative to  $d_{ij}$ .

This constraint can easily be understood with the geometric representation of an example. Suppose 4 sensors. A plane is defined by three non-collinear points. As such, any set of 3 sensors  $x_1, x_2$  and  $x_3$  will define a plane  $u, v$  and their distance matrix will surface 2 positive eigenvalues. Adding a fourth sensor  $x_4$  introduces two possibilities: the additional sensor will either be on the plane (or close to it), or away from the plane (thus indicating  $n = 2$  or  $n = 3$ , and thus a three dimensional space  $u, v, w$ ) as shown in Figure 2.



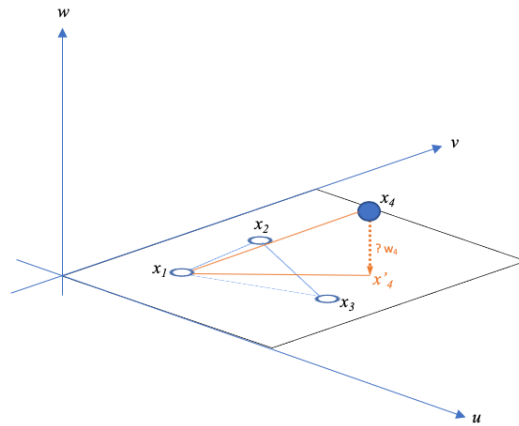


Figure 2. Plane determination from AP sub-sets.

Thus, if we note  $d_{ij}$  the distance between points  $i$  and  $j$ ,  $u_i$  the component of point  $x_i$  along the  $u$  axis, and  $v_i$  and  $w_i$  its components along the  $v$  and  $w$  axis respectively, it is possible that  $w_4 = w_1 = w_2 = w_3 = 0$ , bringing  $x_4$  to the plane defined by  $x_1, x_2$  and  $x_3$ . Quite naturally, in a multi-storey building, it could also happen that  $x_1, x_2$  and  $x_3$  are on different floors, forming a plane that is not parallel to the ground (and  $x_4$  could be yet on a third floor). However, as will be seen below, this scenario does not apply in the case of FTM, as the signal from one sensor would not be detected across two floors.

However, FTM does not provide  $d_{12}, d_{13}, d_{14}, d_{23}, d_{24}$  and  $d_{34}$ , but  $\tilde{d}_{12}, \tilde{d}_{13}, \tilde{d}_{14}, \tilde{d}_{23}, \tilde{d}_{24}$  and  $\tilde{d}_{34}$ . Determining the dimension of the linear span of  $\{x_1, x_2, x_3, x_4\}$  is congruent to computing  $w_4$ . As  $\tilde{d}_{14} = k_{14}d_{14}$ , and therefore, considering a evaluated distance matrix reduced to the pair  $(x_1, x_4)$ , then the matrix  $B_{14}$  obtained after double centering becomes:

$$B_{14} = -\frac{1}{2}J\tilde{D}_{14}^2 = -\frac{1}{2}k_{14}JD_{14}^2 \quad (12)$$

And the simple resolution to U shows that:

$$\begin{aligned} \tilde{d}_{14} &= \|k_{14}d_{14}\|_2 \\ &= k_{14}\|d_{14}\|_2 \\ &= \sqrt{(k_{14}u_4 - k_{14}u_1)^2 + (k_{14}v_4 - k_{14}v_1)^2 + (k_{14}w_4 - k_{14}w_1)^2} \end{aligned} \quad (13)$$

If  $w_1 = 0$  to inscribe  $x_1$  onto the plane defined by  $x_1, x_2, x_3$ , then:

$$k_{14}^2 w_4^2 = k_{14}^2 ((d_{14})^2 - (u_4 - u_1)^2 - (v_4 - v_1)^2) \quad (14)$$

And naturally:

$$w_4^2 = k_{14}^2 d_{14}^2 - d_{14}'^2 = d_{14}'^2 (k - 1) \quad (15)$$

Even if  $x_1, x_2, x_3, x_4 \in \mathbb{R}^2$ , any imprecision in the measurement of  $\tilde{d}_{14}$  generates a dilation  $k_{14}$  that is directly translated into a non-null  $w_4$  proportional to  $k_{14} \geq 1$ . In an environment where vertical separation is often considerably less than horizontal separation, an unknown pairwise  $k_{ij}$  soon prevents the resolution of  $N \in \mathbb{R}^N$ . For example, in a deployment where APs or sensors are positioned every 20 to 25 meters and floors (including slabs and isolation) are 4 meters apart, a measurement of  $\tilde{d}_{14} = 25$  prevents the experimenter from determining if  $d_{14} = 25$  and  $w_4 = 0$  or if  $20 \leq d_{14} \leq 24.68$  and  $4 \leq w_4 \leq 15$ , if  $k_{14}$  cannot deterministically be determined to be 1.2 or lower.

In the case of Wi-Fi ranging, this issue is easily solved with Received Signal Strength Indicator (RSSI) evaluations and LoS path loss equations. But we postulate that time of flight techniques alone cannot solve this issue unless  $k$  is consistent across all  $d_{ij}$  and

estimated precisely. As will be demonstrated below, such quantization and uniformization of  $k$  is possible, but the experimenter cannot know if the dilation is due to a wall or the material separating two floors.

mMDS does introduce the idea of different dilation values with several possible techniques, commonly centered around the idea of defining a cost function, often called the strain or stress function. Such function is defined for example in [19] as

$$\sigma(D) = \sum_{i \neq j \neq k=1}^n w_{ijk} (\epsilon_{ijk} - \tilde{d}_{ij})^2$$

where  $\epsilon_{ijk}$  is an ideal distance set between the points  $i, j$  and  $k$ , given the observed measurement  $\tilde{d}_{ij}$ , and  $w_{ijk}$  is a weight used to express the relative importance or precision of a given measurement  $\tilde{d}_{ij}$ . This relative importance does not aim at resolving dilation asymmetries. In fact, the resolution to  $U$  can take several iterative forms around mechanisms all aiming at minimizing the cost function  $\sigma(D)$ . In all cases, the choice of  $w_{ijk}$  is critical, as it can be different for each  $\tilde{d}_{ij}$ . However, in most cases,  $w_{ijk}$  is modulated based either on confidence indices ( $w_{ijk}$  is high for  $\tilde{d}_{ij}$  known to be close to the ground truth) or on point criticality ( $w_{ijk}$  is high for  $i$  and  $j$  being anchors important to the problem to solve, e.g., connection points to other networks etc.), not on dilation asymmetries. Other MDS techniques proceed with similar logic, with the general idea that distances may be noisy, but the noise being unknown, it can be considered as Gaussian, i.e., symmetric in most directions. Regardless of the method chosen, both the minimization and the double centering techniques tend, as noted in [20], to center the points along the mean of the error, and therefore to center the error. However, in the case of FTM (and probably multiple other time-of-flight-based distance estimation techniques), the noise is not Gaussian throughout the matrix.

#### 4.1.3. Error Averaging

This last point causes the third additional difficulty. mMDS only considers the positive eigenvalues. This is a necessary requirement of the distance-to-coordinate resolution process, which needs the coordinate  $U$  to be derived from a centered matrix  $B$  eigendecomposition ( $B = Q\Lambda Q'$  described in the previous section). Through this process, the coordinate matrix  $U$  is found as  $U = Q\Lambda^{1/2}$ . To maintain  $\Lambda$  in  $R$  (i.e., ensure that such matrix does not have a complex part), and working on the principle that the number of eigenvalues reflects the object dimension,  $(\Lambda^{1/2})^2$  needs to have an expression solely in  $R$  (i.e., no complex part). This requirement means that negative eigenvalues are ignored.

However, as displayed in Table 1, real world FTM measurements typically produce through the mMDS process multiple large positive eigenvalues, but also, sometimes large, negative eigenvalues. As demonstrated in [20],  $B$  is an approximation of  $XX'$ , because it is built from  $\tilde{D}$ , not from  $D$ . As a consequence, writing  $R = [r_{ij}] = B - XX'$  and  $\|R\|_F^2$  the square of  $R$  Frobenius norm, and  $tr(R)$  its trace, [20] shows that the resulting coordinates estimation error can be found to be:

$$L(X) = 2n \sum_{i=1}^n r_{ii}^2 + 2(tr(R))^2 + 4\|R\|_F^2 \quad (16)$$

This result shows that the error increases as the difference between  $d_{ij}$  and  $\tilde{d}_{ij}$  increases. It is also easily seen that the mMDS process centers the error. This issue can easily be seen through a simple example. Suppose a simple measured distance matrix with 3 points, each distance being affected by a specific  $k_{ij}$  dilation factor:

$$\tilde{d}_3 = \begin{bmatrix} 0 & k_{12}d_{12} & k_{13}d_{13} \\ k_{12}d_{12} & 0 & k_{23}d_{23} \\ k_{13}d_{13} & k_{23}d_{23} & 0 \end{bmatrix} \quad (17)$$

**Table 1.** Building 1—Matrix of ground truth distances and related eigenvalues.

Device	1	2	3	4	5	6	7	8	9	10	11	12	13	Eigenvalues
1	0	2086	4183	5632	1430	3655	5262	2452	3275	4387	3295	5564	6702	$5.6797 \times 10^7$
2	2086	0	2135	3581	1704	1999	3414	3734	2895	3187	3722	4242	5150	$2.1831 \times 10^7$
3	4183	2135	0	1470	3362	1312	1564	5278	3405	2582	4672	3137	3579	42,674
4	5632	3581	1470	0	4789	2475	1442	6650	4543	3310	5875	3352	3221	34,379
5	1430	1704	3362	4789	0	2507	4135	2045	1875	3007	2200	4182	5375	28,945
6	3655	1999	1312	2475	2507	0	1623	4218	2118	1407	3422	2275	3156	27,480
7	5262	3414	1564	1442	4135	1623	0	5749	3404	1978	4750	1920	2033	21,641
8	2452	3734	5278	6650	2045	4218	5749	0	2504	4118	1530	5172	6490	5206.5
9	3275	2895	3405	4543	1875	2118	3404	2504	0	1605	1345	2670	3987	$5.8021 \times 10^{-9}$
10	4387	3187	2582	3310	3007	1407	1978	4118	1605	0	2892	1175	2415	−10,435
11	3295	3722	4672	5875	2200	3422	4750	1530	1345	2892	0	3825	5153	−11,658
12	5564	4242	3137	3352	4182	2275	1920	5172	2670	1175	3825	0	1318	−46,289
13	6702	5150	3579	3221	5375	3156	2033	6490	3987	2415	5153	1318	0	$-1.1762 \times 10^5$

In this simple configuration:

$$J = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix} \quad (18)$$

And thus:

$$B = -\frac{1}{16} \begin{bmatrix} -3k_{12}^2d_{12}^2 - 3k_{13}^2d_{13}^2 + k_{23}^2d_{23}^2 & 5k_{12}^2d_{12}^2 - k_{13}^2d_{13}^2 - k_{23}^2d_{23}^2 & -k_{12}^2d_{12}^2 + 5k_{13}^2d_{13}^2 - k_{23}^2d_{23}^2 \\ 5k_{12}^2d_{12}^2 - k_{13}^2d_{13}^2 - k_{23}^2d_{23}^2 & -3k_{12}^2d_{12}^2 + k_{13}^2d_{13}^2 - 3k_{23}^2d_{23}^2 & -k_{12}^2d_{12}^2 - k_{13}^2d_{13}^2 + 5k_{23}^2d_{23}^2 \\ -k_{12}^2d_{12}^2 + 5k_{13}^2d_{13}^2 - k_{23}^2d_{23}^2 & -k_{12}^2d_{12}^2 - k_{13}^2d_{13}^2 + 5k_{23}^2d_{23}^2 & k_{12}^2d_{12}^2 - 3k_{13}^2d_{13}^2 - 3k_{23}^2d_{23}^2 \end{bmatrix} \quad (19)$$

From  $B$ , it can clearly be seen that dilation factors that affect a single distance pair in  $\tilde{D}$  are projected as weights for the computation of the centered matrix  $B$ , affecting all other entries of  $B$  in the process. The weights, because of the structure of  $J$ , affect differently the various pairs. The sum of the contribution of each dilation factor  $k_{ij}$  in  $B$  is now 1 for each row or column where the factor was present in  $\tilde{D}$  and  $-1$  for each row or column where the factor was not present. Overall, the contribution in  $B$  is reflective of the overall contribution in  $\tilde{D}$ . However, centering the matrix also has the effect of distributing the dilation factors, and thus distributing individual dilation factors.

The outcome of such distribution is that a single large dilation factor will then affect the computed position of all points in  $U$ , thus distributing the error to all positions. In a distance setting where the noise is Gaussian across all sensors, this distribution has only minor effects. However, in settings, like indoor location with FTM, where noise asymmetry is present, and where accuracy depends on identifying the RF interference localities and the associated dilation differences between pairs, mMDS, used alone, can provide an acceptable result, but that will always be worse than quality LoS measurements. As a consequence, highly dilated distance pairs get hidden in the transformation, and precise measurements get degraded by the contribution of dilated pairs.

Therefore, there is a need for a method that can identify and compensate for the highly dilated segments, to attempt to reduce their dilation before they are injected in a method, such as mMDS, where all segment contributions are treated equally.

## 5. Materials and Methods

### 5.1. First Component: Wall Remover-Minimization of Asymmetric Errors

One first contribution of this paper is a method to reduce dilation asymmetry. Space dimension is resolved using other techniques (e.g., RSSI-based), and Section 6 will provide an example. Once the dimension space has been reduced to  $\mathbb{R}^2$  and only sensors on the same floor, we want to reduce the dilation asymmetry. Figure 1 illustrates a typical asymmetry scenario. In this simplified representation, a strong obstacle appears between the sensor  $x_1$  and sensors  $x_6, x_7$  and  $x_8$ , causing the distances  $d(x_6, x_1)$ ,  $d(x_7, x_1)$  and  $d(x_8, x_1)$  to be appear as  $d(x_6, x'_1)$ ,  $d(x_7, x'_1)$  and  $d(x_8, x'''_1)$  respectively. At the same time, supposing that the system is calibrated properly and LoS conditions exist elsewhere, the obstacle does not affect distance measurements between  $x_1$  and  $x_2, x_3, x_4, x_5$  and  $x_9$ , that are all approximated along the same linear dilation factor  $k$ .

In such case, using a centering technique to average the error may make sense algebraically, but not geometrically. Finding a method to reduce the error while detecting its directionality is therefore highly desirable. Luckily, geometry provides great methods to this mean, that only have the inconvenience of requiring multiple comparisons. Such process may be difficult when performing individual and mobile station ranging, but becomes accessible when all sensors are static. When implemented in a learning machine, this method can be implemented at scale and reduce the distance error only when it is directional, thus outputting a matrix  $\tilde{D}$  which  $k$  factor is closer to uniformity. It should be noted that the purpose of such method should not be to fully solve the MDS problem, as some pair-distances are usually not known, and the method has a limited scope (i.e., it cannot assess some of the pairs, as will be seen below). However, in many cases, sensors can be found that display interesting properties displayed on the left part of Figure 3. In this scenario, 3 sensors  $x_1, x_2$  and  $x_3$  are selected that form a triangle. The triangle represented in the left part of Figure 3 is scalene, but the same principle applies to any triangle. A fourth sensor  $x_4$  is found which distance to  $x_1, x_2$  and  $x_3$  is less than  $d_{12}, d_{13}$  or  $d_{23}$ , thus placing  $x_4$  within the triangle formed by  $(x_1, x_2, x_3)$ .

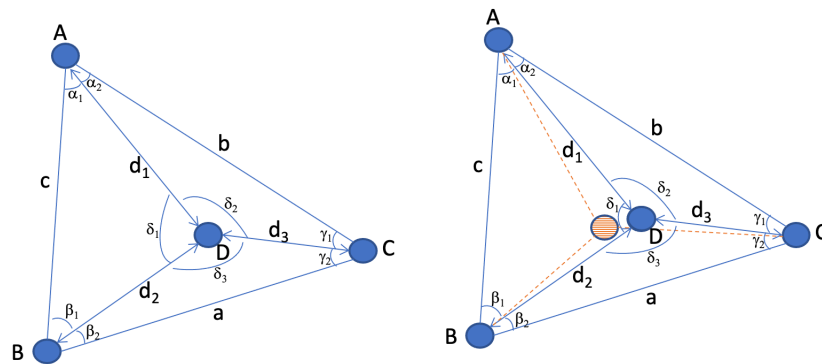


Figure 3. Sensor geometric relationship.

A natural property of such configuration is that  $\delta_1 + \delta_2 + \delta_3 = 2\pi$  and, in any of the triangles  $(x_1, x_2, x_4)$ ,  $(x_1, x_3, x_4)$  or  $(x_2, x_3, x_4)$ , one side can be expressed as a combination of the other two and of its opposite angle. For example,  $d_{24}$  in  $(x_1, x_2, x_4)$  can be expressed as:

$$d_{24}^2 = d_{12}^2 + d_{14}^2 - 2d_{12}d_{14}\cos(\alpha_1) \quad (20)$$

The above easily allows us to find  $d_{24}$  and can also be used to determine angles from known distances, for example  $\delta_1$ , knowing that:

$$\cos(\delta_1) = \frac{d_{14}^2 + d_{24}^2 - d_{12}^2}{2d_{14}d_{24}} \quad (21)$$

Therefore, angles and missing distances can be found from known distances. In an ideal world, properties (20) and (21) are verified for each observed triangle  $((x_1, x_2, x_3), (x_1, x_2, x_4), (x_2, x_3, x_4)$  and  $(x_1, x_3, x_4))$ . In a noisy measurement scenario, inconsistencies are found. For example, an evaluation of the triangle  $(x_1, x_2, x_4)$  may be consistent with the left side of Figure 3, but an evaluation of triangle  $(x_1, x_3, x_4)$  may position  $x_4$  in the hashed representation  $x'_4$  of the right part of Figure 3. It would be algebraically tempting to resolve  $x_4$  as the middle point between both possibilities (mean error). However, this is not the best solution. In Figure 3 simplified example, the most probable reason for the inconsistency is the presence of an obstacle or reflection source between  $x_3$  and  $x_4$ . If all distances are approximations, some of them being close to the ground truth and some of them displaying a larger  $k$  factor, a congruent representation of such asymmetry is that  $k$  is larger for  $x_4x_3$  than it is for the other segments. Quite obviously, other possibilities can be found. For example, in the individual triangle  $(x_1, x_2, x_4)$ , it is possible that  $k$  is larger for the segments  $d_{12}$  than it is for segments  $d_{24}$  and  $d_{14}$ . However, as  $x_4x_3$  is compared to other segments and the same anomaly repeats, the probability that the cause is an excessive stretch on  $x_4x_3$  increases.

As the same measurements are performed for more points, the same type of inconsistency appears for other segments. Thus an efficient resolution method is to identify these inconsistencies, determine that the distance surfaced for the affected segment is larger than a LoS measurement would estimate, then attempt to individually and progressively reduce the distance (by a local contraction factor that we call  $0 < \zeta_{ij} \leq 1$ ), until inconsistencies are reduced to an acceptable level (that we call  $\gamma$ ).

Thus, formally, a learning machine that we call a geometric Wall Remover engine, is fed with all possible individual distances in the matrix, and compares all possible iterations of sensors forming a triangle and also containing another sensor. Distance matrices do not have orientation properties. When considering  $(x_1, x_4, x_3)$  for example,  $x_4$  can be found on either side of segment  $x_1x_3$ , and the solution can also be a triangle in any orientation. However, adding a fourth point  $(x_2)$ , which distance to  $x_4$  is evaluated, can constrain  $x_4$  within the  $(x_1, x_2, x_3)$  triangle. As the evaluation proceeds iteratively throughout all possible triangles that can be formed from the distance matrix, the system first learns to place the sensors relative to each other. The resulting sensor constellation can have globally any orientation, but the contributing points partially appear in their correct relative position.

Then, each time a scenario matching the right side of Figure 3 is found, the algorithm learns the asymmetry and increases the weight  $w$  of the probability  $p$  that the matching segment ( $x_3x_4$  in this example) has a dilated  $k_{ij}$  factor (cf. Algorithm 1). As a segment may be evaluated against many others, its  $w$  may accordingly increase several times at each run. The algorithm starts by evaluating the largest found triangles first (sorted distances from large to small), because they are the most likely to be edge sensors. This order is important, because in Figure 3 example, a stretched  $x_2x_4$  value may cause  $x_4$  to be graphed on the right side of segment  $x_1x_3$ , and thus outside of  $(x_1, x_2, x_3)$ . This risk is mitigated if multiple other triangles within  $(x_1, x_2, x_3)$  can also be evaluated. An example is depicted by Figure 4. In this simplified representation (not all segments are marked), the position of  $x_4$  is constrained by first evaluating  $(x_5, x_4, x_7)$ ,  $(x_5, x_4, x_{10})$ ,  $(x_{10}, x_4, x_{12})$  and  $(x_7, x_4, x_{12})$  against  $(x_5, x_{10}, x_7)$ ,  $(x_5, x_{10}, x_{12})$ ,  $(x_5, x_7, x_{12})$  and  $(x_{10}, x_7, x_{12})$ . This evaluation allows the system to surface the high probability of the stretch of segment  $x_4x_7$ , suggesting that  $x_4$  should be closer to  $x_7$  than the measured distance  $\tilde{d}_{x_4x_7}$  suggests, but not to the point of being on the right side of  $x_1x_3$ . The system can similarly detect a stretch between points  $x_3$  and  $x_8$  (but not between  $x_7$  and  $x_8$  or  $x_{10}$  and  $x_3$ ).

**Algorithm 1:** Wall Remover Algorithm

---

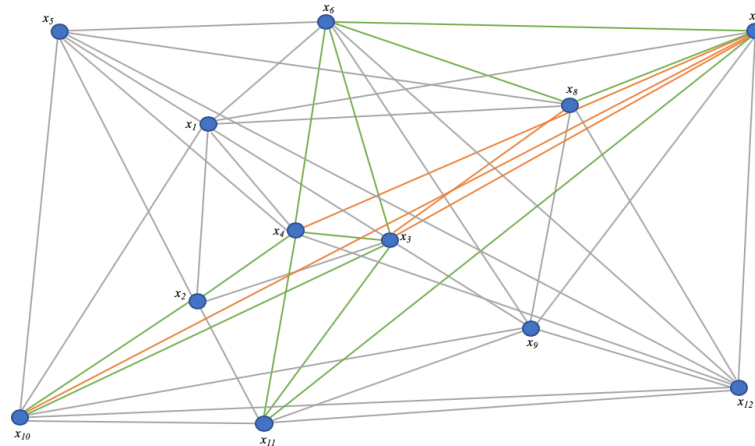
**Input** :  $\epsilon$ : learning rate  
 $\gamma$ : acceptable error range  
**Output**: optimized  $d_{ij}$

```

1 // convert  $\tilde{D} \in \mathbb{R}^2$  to pairwise distance list  $d_{ij}$ 
2  $d_{ij}[i, j, dist_{ij}] \leftarrow dist2list(\tilde{D})$ ;
3 // set a list of same length  $\zeta_{ij}$  of all ones
4 // set a list of same length  $w_{ij}$  of all ones
5  $d_{ij}[i, j, dist_{ij}, \zeta_{ij}, w_{ij}] := \{d_{ij}, \zeta_{ij}, w_{ij}\}$ ;
6 while true do
7   reverse_sort  $d_{ij}$ ;
8   for  $d_{ij} \geq d_{ik} > d_{jk} > 0$  do
9     find  $l : d_{il}, d_{jl}, d_{kl} \neq 0$ ;
10    compute  $\widehat{ijk}, \widehat{jki}, \widehat{kij}, \widehat{ijl}, \widehat{jkl}, \widehat{kil}$ ;
11    if  $\widehat{ijl} > \widehat{ijk} \vee \widehat{jkl} > \widehat{jki} \vee \widehat{kil} > \widehat{kij}$  then
12      find next  $l$  ( $l$  is outside  $(i, j, k)$ );
13    if  $d_{il} / \sqrt{d_{ij}^2 + d_{jl}^2 - 2d_{ij}d_{jl}\cos(\widehat{ijk} - \widehat{kjl})} > 1$  then
14       $w_{il} := w_{il} + (d_{il} / \sqrt{d_{ij}^2 + d_{jl}^2 - 2d_{ij}d_{jl}\cos(\widehat{ijk} - \widehat{kjl})} - 1)$ ;
15      repeat along  $d_{jl}$  and  $d_{kl}$ ;
16    next  $l$ ;
17   foreach  $d_{ij}$  do
18      $p_{ij} := 1 / (1 + e^{w_{ij}})$ ; // compute stretch probability:
19   reverse_sort  $p_{ij}$ ;
20    $tp_{ij} \leftarrow \text{pick\_top } p_{ij}$ ;
21   if  $tp_{ij} < \gamma$  then
22     break;
23    $d_{ij} := d_{ij}\epsilon$ ;
24    $\zeta_{ij} := \zeta_{ij} - \epsilon$ ;

```

---



**Figure 4.** Stretch detection between  $x_3$  and  $x_8$ .

At the end of the first training iteration, the system outputs a sorted list of segments with the largest stretch probabilities (largest  $w$  and therefore largest  $p$ ). The system then picks that segment with that largest stretch, and attempts to reduce its stretch by proposing a contraction of the distance by an individual  $\zeta_{ij}$  factor. The  $\zeta_{ij}$  factor can be a step increment (similar to other machine learning algorithms *learning rate* logic) or can be proportional to the stretch probability. After applying the contraction, the system runs the next iteration, evaluating progressively from outside in, each possible triangle combination. The system



then recomputes the stretch probabilities and proposes the next contraction cycle. In other words, by examining all possible triangle combinations, the system learns stretched segments and attempts to reduce the stretch by progressively applying contractions until inner neighboring angles become as coherent as possible, i.e., until the largest stretch is within a (configurable) acceptable range  $\gamma$  from the others.

This method has the merit of surfacing points internal to the constellation that display large  $k$  factors, but is also limited in scope and intent. In particular, it cannot determine large  $k$  factors for outer segments, as the matching points cannot be inserted within triangles formed by other points. However, its purpose is to limit the effect on measurements of asymmetric obstacles or sources of reflection.

## 5.2. Second Component: Iterative Algebro-Geometric EDM Resolution

The output of the wall remover method is a matrix with lower variability to the dilation factor  $k$ , but the method does not provide a solution for an incomplete and noisy EDM. Such reduction still limits the asymmetry of the noise, which will therefore also limit the error and its locality when resolving the EDM, as will be shown below. Noise reduction can be used on its own as a preparatory step to classical EDM resolution techniques. It can also be used in combination with the iterative method we propose in this section, although the iterative method has the advantage of also surfacing dilation asymmetries, and thus could be used directly (without prior dilation reduction). Combined together, these two techniques provide better result than standard EDM techniques.

EDM resolution can borrow from many techniques, which address two contiguous but discrete problems: matrix completion and matrix resolution. In most cases, the measured distance matrix  $\tilde{D}$  has missing entries, indicating unobserved distances. In the case of FTM, these missing distances represent out-of-range pairs (e.g., sensors at opposite positions on a floor or separated by a strong obstacle, and that cannot detect each other). A first task is to complete the matrix, by estimating these missing distances. Once the matrix contains non-zero numerical values (except for the diagonal, which is always the distance  $d_{ii}$  and therefore always 0), the next task is to reconcile the inconsistencies and find the best possible distance combination.

Several methods solve both problems with the same algorithm and [21] provides a description of the most popular implementations. We propose a geometric method, that we call Iterative Algebro-Geometric EDM resolver (Algorithm 2), which uses partial matrix resolution as a way to project sensor positions geometrically onto a  $\mathbb{R}^2$  plane, then a mean cluster error method to identify individual points in individual sets that display large asymmetric distortions (and should therefore be voted out from the matrix reconstruction). By iteratively attempting to determine and graph the position of all possible matrices for which point distances are available, then by discarding the poor (point pairs, matrices) performers and recomputing positions without them, then by finding the position of the resulting position clusters, the system reduces asymmetries and computes the most likely position for each sensor.

We want to graph the position of each point in the sub-matrix. A pivot  $x_1$  is chosen iteratively in  $\tilde{d}_m$ . In the first iteration,  $x_1 = i_1$ , then  $x_1 = i_2$  in the second iteration, and  $x_1 = i_m$  in the last iteration. For each iteration,  $x_1$  is set as the origin, and  $x_1 = (0, 0)$ . An example is displayed in Figure 5. The next point  $x_2$  in the matrix ( $i_2$ ) is iteratively set along the  $x$ -axis, and  $x_2 = (\tilde{d}_{x_1 x_2}, 0)$ . If  $m > 2$ , then the position of each other point  $x_i$  of the set  $\{x_1, x_2, x_i\}$  is found using standard triangular formula illustrated by the points in Figure 5 and where:

$$u_{x_i} = \frac{x_i x_2^2 - x_1 x_i^2 - x_1 x_2^2}{-2x_1 x_2} \quad (22)$$

and

$$v_{x_i} = \sqrt{x_1 x_i^2 - u_{x_i}^2} \quad (23)$$

**Algorithm 2:** Iterative algebro-geometric EDM resolution algorithm

---

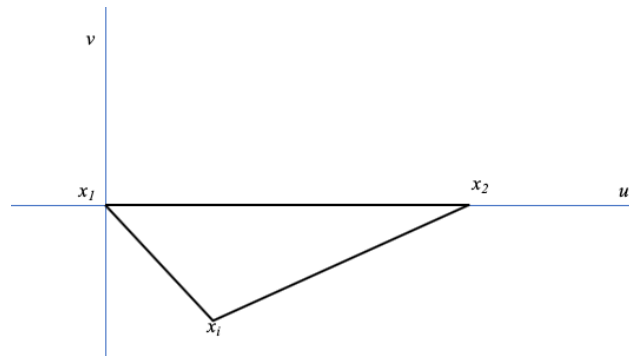
**Input:**  $\tilde{D}_{m \times m} \in \mathbb{R}^2$

```

1 for  $i$  from 1 to  $m - 1$ : do
2    $\lfloor$  generate all  $\tilde{D}_{i \times (i+1)}$  to  $\tilde{D}_{i \times (m-i)}$  matrices;
3   remove each  $d_{ij} = 0$  from each matrix;
4   remove any  $\tilde{D}_{1 \times 1}$  matrix;
5   for generated  $\tilde{D}_{n \times n}$  do
6      $p = q = 1$ ;
7     for  $o = 1$  to  $n$  do
8       plot  $x_o = (0, 0)$ ;
9       plot  $x_{o+p} = (d_{op}, 0)$ ;
10       $x_{o+q} = (u_{o+q}, v_{o+q})$ ;
11       $u_{o+q} = (\frac{x_{o+q}x_{o+p}^2 - x_o x_{o+q}^2 - x_o x_{o+p}^2}{-2x_o x_{o+p}}, )$ ;
12       $v_{o+q} = (\sqrt{x_o x_{o+q}^2 - u_{o+q}^2})$ ;
13      if  $d_{x_{o+q}-1 x_{o+q}} > v_{o+q}$  then
14         $v_{o+q} := -v_{o+q}$ ;
15      next  $q$ ;
16      next  $p$ , compute  $\widehat{p_{-1} o p}$  and rotate new plot onto previous plot coordinates;
17      next  $o$ , compute  $d_{o-1 o}$  and translate new coordinates into previous reference;
18 for each point do
19    $\lfloor$  compute cluster center position;
20 for each cluster do
21    $\lfloor$  compute  $\sigma$ ;
22 for all clusters do
23    $\lfloor$  compute average  $\sigma$ ;
24 record this initial  $\sigma$ ;
25 for each cluster do
26    $\lfloor$  record any point  $2\sigma$  or more from cluster center;
27 select farthest point of all;
28 if point was an anchor ( $o$  or  $p$ ) then
29    $\lfloor$  mark matching matrix as invalid;
30 else
31    $\lfloor$  select next farthest point;
32 go back to plotting all remaining matrices;
33 when no anchor is outlier anymore, remove all non-anchor outliers;
34 compute cluster means for remaining points;

```

---



**Figure 5.** Geometric representation of a set of 3 points in a submatrix.

Formally, the measured distance matrix  $\tilde{D}$  of  $n$  sensor distances is separated in sub-matrices. Each sub-matrix  $\tilde{d}_m$  contains  $2 \leq m \leq n$  points for which inter-distances were measured, so that:

$$\forall 2 \leq m \leq n \text{ and } \forall i, j \in \tilde{d}_m, \tilde{d}_{ij} > 0 \quad (24)$$

Such formula fixes the position of  $x_i$  above the  $u$ -axis (because  $v_{x_i}$  is always positive). This may be  $x_i$ 's correct position in the sensor constellation in some cases, but can also result in an incorrect representation as soon as the next point  $x_j$  is introduced in the graph. A simple determination of the respective positions of  $x_i$  and  $x_j$  can be made by evaluating their distances, as in the wall remover method. In short, if  $d_{x_i x_j} > v_{x_i}$ , then  $x_i$  and  $x_j$  are on opposite sides of the  $u$ -axis and  $v_{c_j}$  becomes  $-v_{c_j}$ .

As the process completes within the first matrix, the position of  $m$  points are determined using the first pair of points  $x_1$  and  $x_2$  as the reference. In the next iteration,  $x_1$  is kept but  $x_2$  changes from  $i_2$  to  $i_3$ . Algebraically, the first iteration determines the positions based on the pairwise distances expressed in the first 2 columns of the distance matrix, while the second iteration determines the positions based on the pairwise distances expressed in the first and third columns. Both results overlap for the first 3 points  $x_1, x_2$  and  $x_3$ , but not for the subsequent points. This can be easily understood with an example. Recall that for any pair of points  $i$  and  $j$ ,  $d_{ij} = d_{ji}$ . Using  $d_{ij}$  as a representation of both  $d_{ij}$  and  $d_{ji}$ , and the following small matrix of 5 points as an example:

$$\tilde{D}_5 = \begin{bmatrix} 0 & \tilde{d}_{12} & \tilde{d}_{13} & \tilde{d}_{14} & \tilde{d}_{15} \\ \tilde{d}_{12} & 0 & \tilde{d}_{23} & \tilde{d}_{24} & \tilde{d}_{25} \\ \tilde{d}_{13} & \tilde{d}_{23} & 0 & \tilde{d}_{34} & \tilde{d}_{35} \\ \tilde{d}_{14} & \tilde{d}_{24} & \tilde{d}_{34} & 0 & \tilde{d}_{45} \\ \tilde{d}_{15} & \tilde{d}_{25} & \tilde{d}_{35} & \tilde{d}_{45} & 0 \end{bmatrix} \quad (25)$$

The first iteration ignores  $\tilde{d}_{34}$  and  $\tilde{d}_{35}$  that are represented in the second iteration (but the second iteration does not represent  $\tilde{d}_{24}$  or  $\tilde{d}_{25}$ ).

As in the second iteration  $i_3$  is used as a reference point for the  $x$  axis, the geometrical representations of the first and the second iterations are misaligned. However,  $x_1$  is at the origin in both cases, and  $x_2$  is represented in both graphs (we note them  $x_2$  and  $x'_2$ ). Using Equation (22), finding the angle ( $\alpha$ ) formed by the points  $x_2 x_1 x'_2$  is straightforward, and projecting the second matrix into the same coordinate reference as the first matrix is a simple rotation of the second matrix, defined by  $T$  as:

In the subsequent iterations,  $x_1$  ceases to be at the origin. Depending on the sub-matrix and the iteration,  $x_1$  may or may not be in the new matrix. However, 2 points  $i = (u_i, v_i)$  and  $j = (u_j, v_j)$  can always be found that are common to both the previous matrix  $\tilde{d}_p$  and the next matrix  $\tilde{d}_n$ . Projecting  $\tilde{d}_n$  into the same coordinate reference as  $\tilde{d}_p$  is here again trivial, by first moving the coordinates of each point found from  $\tilde{d}_n$  by  $proj(u_i) = u_{ip_n} - u_{ip_p}$  and  $proj(v_i) = v_{ip_n} - v_{ip_p}$  then perform a rotation using Equation (22). These operations are conducted iteratively. As measured distances are noisy, for each point represented in  $\tilde{D}$ , different coordinates appear at the end of each iteration, thus surfacing for each point a cluster of computed coordinates. Figure 6 represents this outcome after 4 iterations over a 5-point matrix and  $x_1(0,0)$  fixed about the first point.

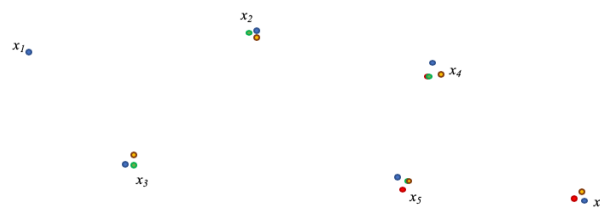


Figure 6. Graphing points in a  $5 \times 5$  matrix using the geometric approach.

The effect of projection and rotation makes that the red and green points are overlapping in  $x_4$ , and the green and orange point are overlapping in  $x_5$ . The choice to determine which points should be used to project  $\tilde{d}_n$  onto  $\tilde{d}_p$  is sequentially obvious, but arbitrary otherwise. In the example displayed in Figure 6,  $x_6$  instead of  $x_2$  could have been chosen to project iteration green onto iteration red etc. A reasonable approach could be to compute iteratively, another one to compute all possible combinations, a third approach is to compute the mean of the positions determined for a point as the best representation of the likely position of a given point for a given iteration. The second method is obviously more computationally intensive, but provides a higher precision in the final outcome. As a cluster of positions appears for each point, representing the computed position of each sensor, asymmetries and anomalies can here as well be surfaced. All points associated with a sensor form a cluster, which center can be determined by a simple mean calculation, where for each cluster center  $\mu_i = (x_{\mu_i}, y_{\mu_i})$  and  $m$  associated points:

$$u_{\mu_i} = \frac{\sum_{j=1}^m u_j}{m} \quad (26)$$

And

$$v_{\mu_i} = \frac{\sum_{j=1}^m v_j}{m} \quad (27)$$

$$T = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (28)$$

Projections that are congruent will display points that are close to one another for a given cluster. However, the graph will also display some representations that display a large deviation. This deviation can be asymmetric. It is then caused by a dilation factor  $k$  different for a sensor pair than for the others. In most cases, strong obstacles or reflection sources may increase the dilation factor. The geometric wall remover method exposed in the previous section of this work is intended to reduce such effect. As the algorithm acts on the angles of adjacent triangles, it is more precise than this section of our proposed method. However, it may happen that the dilation occurs among pairs than the geometric wall remover cannot identify (for example because the pair is formed with sensors at the edge of the constellation).

The deviation also surfaces matrices coherence. A coherent matrix contains a set of distances displaying a similar dilation factor  $k$ . An incoherent matrix contains one or more distance displaying a  $k$  factor largely above or below the others. For example, several sensors may be separated from each other by walls, but be in LoS of a common sensor, which will display a  $k$  factor smaller than the others. Such sensor is an efficient anchor, i.e., an interesting point  $x_1(0, 0)$  or  $x_2(\tilde{d}_{x_1x_2}, 0)$  for the next iteration. On the other hand, some sensors may be positioned in a challenging location and display large inconsistencies when ranging against multiple other sensors. The position of these sensors needs to be estimated, but they are poor anchors for any iteration.

An additional step is therefore to identify good, medium and poor anchors and discard all distances that were computed using poor anchors. The same step can identify and remove outliers pairwise computed positions that deviate too widely from the other computed positions for the same points), and thus accelerate convergence. For each cluster  $i$  of  $m$  points for a given sensor, each at an Euclidian distance  $x_i\mu_i$  from the cluster center  $\mu_i$ , a mean distance to the cluster center can be expressed as:

$$r_i = \frac{\sum_{j=1}^m |x_j - \mu_i|}{m} \quad (29)$$

where  $r_i$  thus expresses the mean radius of the cluster. By comparing radii between clusters, points displaying large  $r_i$  values are surfaced. Different comparison techniques can be used for such comparison. We found efficient the straightforward method of using the

$2\sigma$  rule, where a cluster which radius is more than 2 standard deviations larger than all clusters mean radii is highlighted as an outlier. The associated sensor therefore displays unusually large noise in its distance measurement to the other sensors, and is therefore a poor anchor. Matrices using this sensor as an anchor are removed from the batch and clusters are recomputed without these sensors' contribution as anchors. As the computation completes, each cluster center is used as the best estimate of the associated sensor position. By reducing the variance of the dilation factor  $k$ , by removing sub-matrices and sensor pairs that bring poor accuracy contribution, this method outperforms standard EDM completion methods when  $\mathbb{R}^n$  is known, because it reduces asymmetries before computing positions, but also because the geometric method tends to rotate the asymmetries as multiple small  $3 \times 3$  matrices are evaluated, thus centering the asymmetries around the sensor most likely position.

## 6. Results

### 6.1. Experiment Methodology

We tested this method in five different buildings already equipped with Wi-Fi APs providing active coverage. As the active APs do not support FTM, we position FTM devices near each active AP (typically 15 cm, of half a foot, from each active AP position). Building 1 is a three-storey office building with cubicle areas alternating with blocks of small offices. Our testbed is installed on the second floor. Building 1 is interesting, because the wall structure is irregular, causing different reflection and absorption patterns for each AP pair. The test floor is already equipped with 13 access points positioned at ceiling level. We therefore positioned 13 FTM stations at ceiling level, one near each AP, as represented in Figure 7.

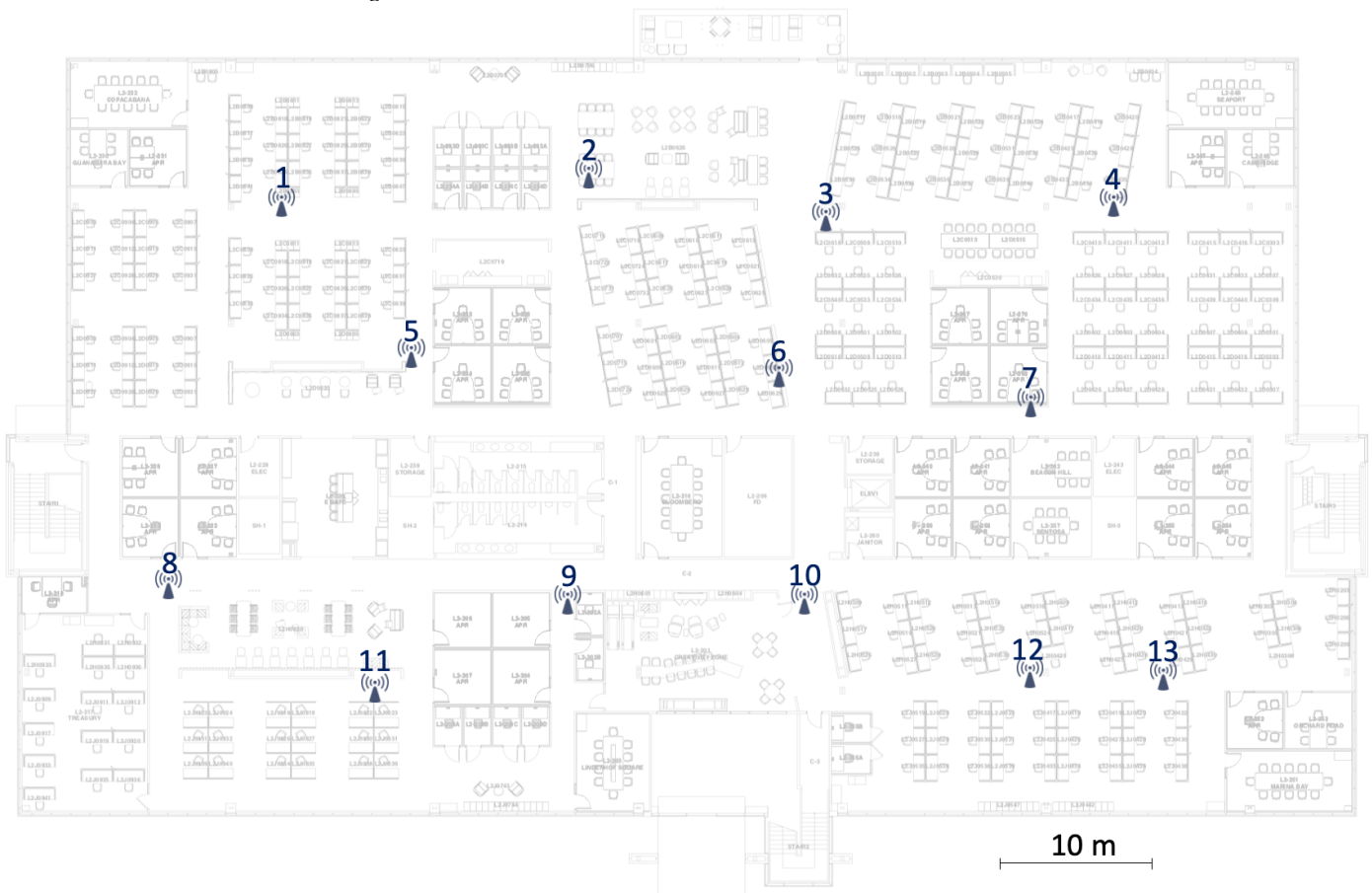
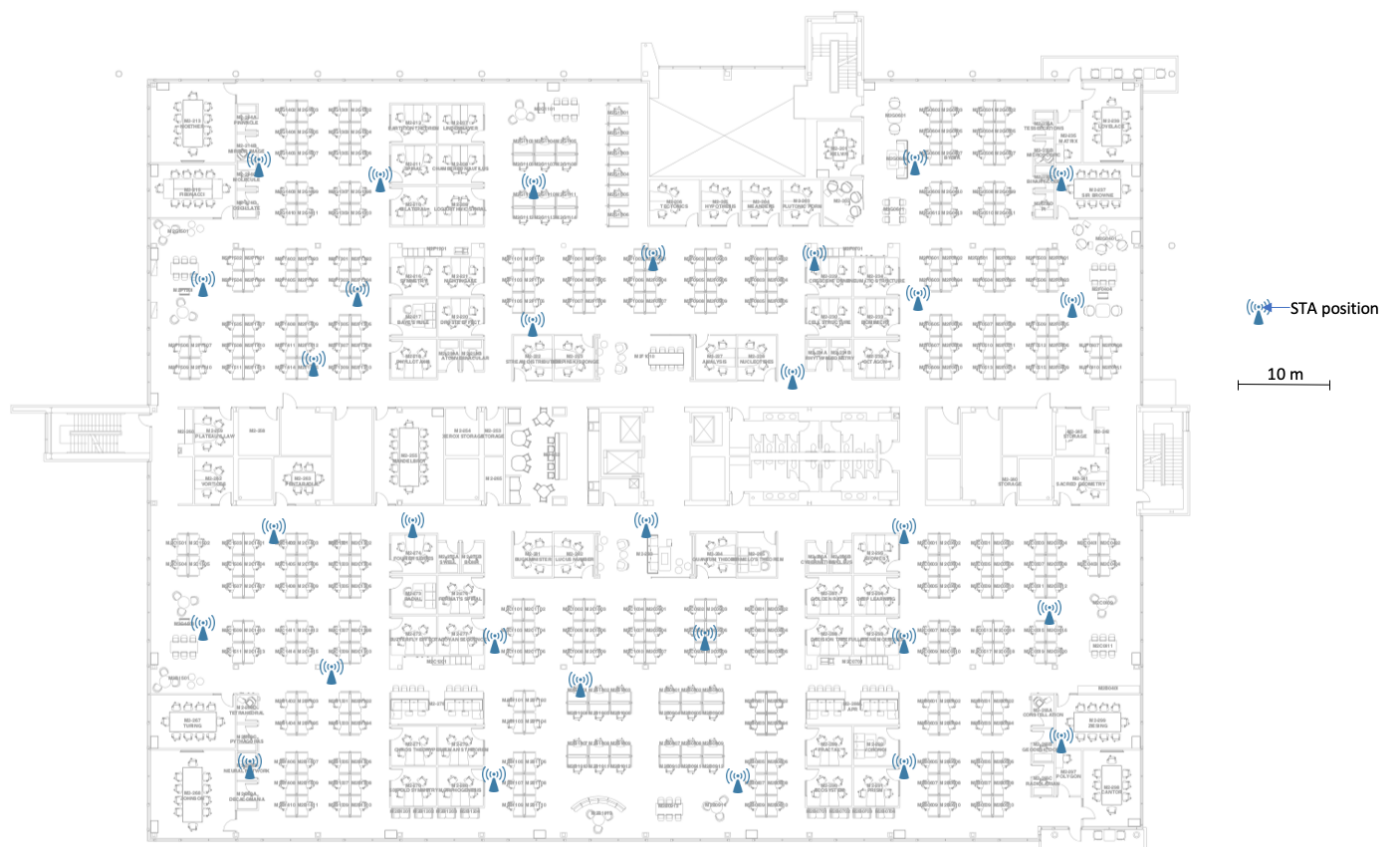


Figure 7. Building 1 experimental setup.

Building 2 is also an office building with 30 APs, with denser sitting density than building 1 and where zones of open space alternate with strong obstacles. Similarly to building 1, one FTM device is positioned 15 cm (1/2 foot) away from each existing AP, as represented in Figure 8.



**Figure 8.** Building 2 experimental setup.

Building 3 is a large open space building with partial high ceilings, used as an enterprise restaurant. It presents high peaks of user and Wi-Fi traffic alternating with no activity. This traffic pattern allows us to evaluate the effect of traffic (or lack thereof) on ranging accuracy. Additionally, the building structure makes that AP height is not consistent throughout the floor, thus presenting an interesting geometry. Building 3 is represented in Figure 9. High-ceiling APs are represented by upside down icons. In this setting, all APs are the same model and use omnidirectional antennas. Thus, the only difference between APs is their height, 3.2 meters for standard ceiling heights, and 7.3 meters for high ceiling APs. Our FTM devices are positioned near each AP as in the previous buildings.

Building 4 is a warehouse, with metallic racks, high ceilings and some APs mounted with directional antennas. An office space is present in the upper right corner of the building. High ceiling APs use omnidirectional antennas and are positioned on poles hanging from the ceiling, between 6.4 and 8.1 meters above the ground. On the side of the aisles, some APs are mounted at 3.4 meters height and equipped with directional patch antennas, as represented in Figure 10. For our experiments, we use these antennas to evaluate the effect of directionality.





Figure 9. Building 3 experimental setup.

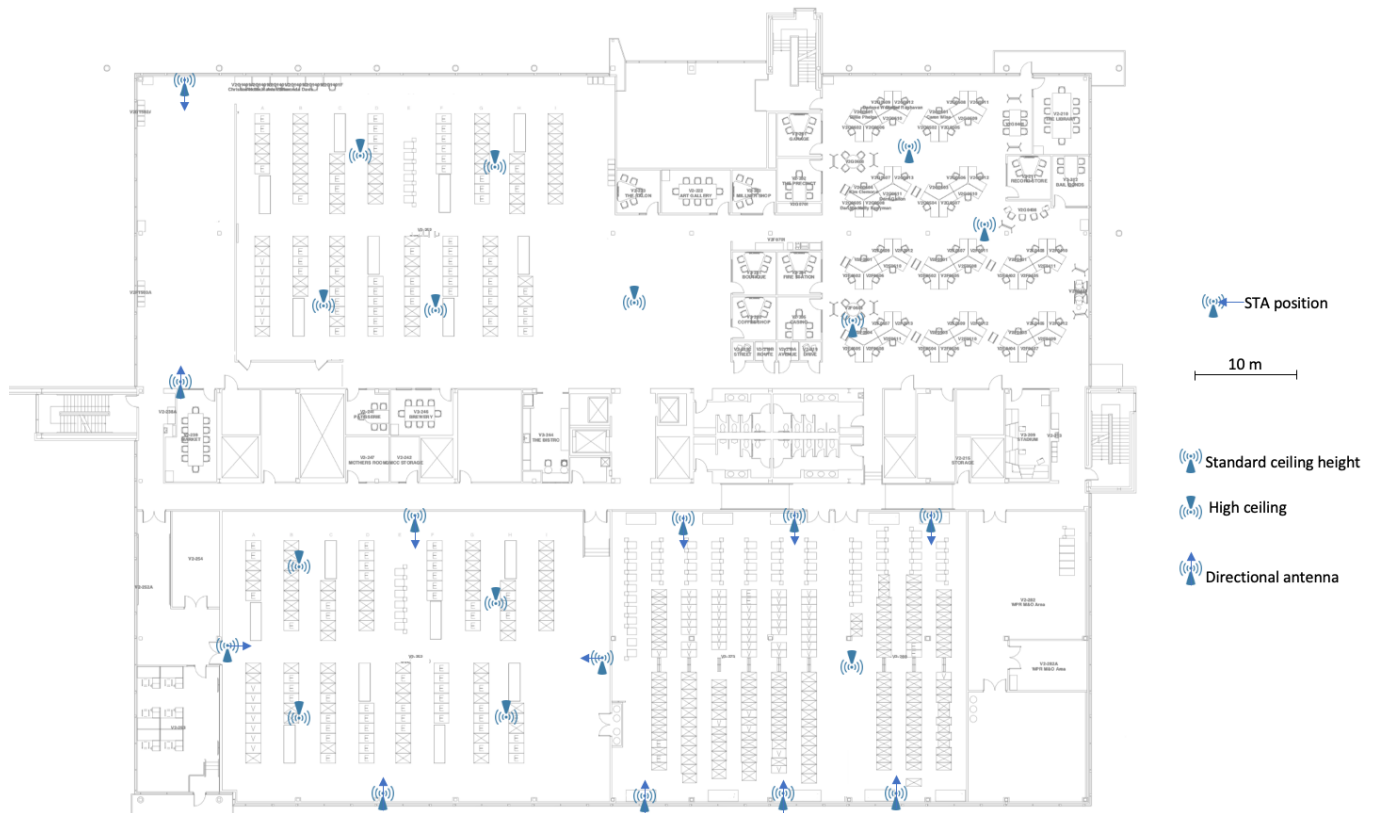


Figure 10. Building 4 experimental setup.

Building 5 is a training center with small to medium-sized classrooms. It presents an interesting denser wall structure (as walls separate the classrooms). As represented in

Figure 11, each classroom has its own AP in the center. The dense (hashed) non-accessible technical area at the center makes that APs on one side do not detect APs on the other side.



Figure 11. Building 5 experimental setup.

For the main presentation of this section, we will focus on building 1 as the main example, and will highlight the other buildings when they present interesting variations. In all cases, our FTM devices are Linux systems equipped with Intel 8260 cards enabled for FTM. In the rest of this section, we will refer to AP-to-AP distances and measurements, although it should be clear that the FTM device associated to each AP was used to operate the measurement. Ground truth distances are known and confirmed with floorplan blueprint and onsite laser ranging. The FTM devices can be configured to act as ISTAs or RSTAs. The system is left active for two days (with the exception of the warehouse, where the system is left active for 6 hours, so as to minimize the disruptions to the warehouse operations). Every hour, the system wakes up, and each station is randomly affected the role of ISTA or RSTA. Each ISTA ranges against the detected RSTAs on various channels for 10 minutes and logs the result. At the end of the collection phase, the logs of all stations are collected and injected into the learning machine.

This duration was chosen to ensure that collection would happen both at quiet times (at night, with no user is in the building) as well as during busy day times. Although daytime measurements do show larger collision and retry counts, no significant difference

could be observed in ranging accuracy. It is therefore likely that the system could only be left active for a shorter duration and yet obtain convergence. Mobile stations are also walked around the building, to determine locations where ranging to indoor APs would be possible. From left entrance and stairs, APs 01, 08 and 11 are reachable. From the main entrance (bottom), APs 11, 09, 10, 12 and 13 are reachable. From the right entrance and stairs, APs 13, 07 and 04 are reachable as can be seen in Figure 12.



Figure 12. Detectability of RSTAs from outside the building.

The purpose of the outdoor mobile stations is to serve as a seed for an initial GPS position communicated to the APs in range of the station. One key determinant is the accuracy of the GPS value seeded by the mobile station. GPS accuracy utilities are installed on different phones. As GPS position is computed, the operator clicks the real location, based on high resolution aerial maps, and the system outputs the error of the computed GPS estimation. As displayed in Figure 13, the error is commonly within half a meter. Inside the building, accuracy collapses (not represented here).

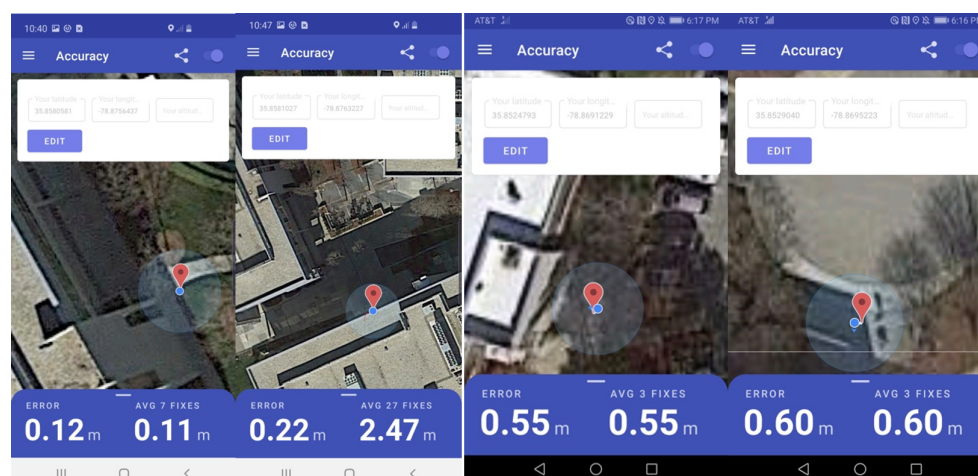


Figure 13. GPS measurement accuracy near the experimental building.

An examination of the distance matrix for all APs (measured through their associated FTM devices) shows high noise. Table 1 displays the ground truth (real distances) and Table 2 the distances obtained via FTM measurements, after applying the  $d_{ij} = d_{ji} =$

$\min\{d_{ij}, d_{ji}\}$  simplification. The ground truth  $13 \times 13$  matrix surfaces 9 positive eigenvalues, because measurements suffered from natural scale and rounding inaccuracies. However, only 2 of these eigenvalues are large, indicating a 2-dimensional geometrical object. The measured FTM matrix, on the other hand, surfaces ‘only’ 7 positive eigenvalues, but all of them are large. The first two eigenvalues are larger by a factor of 10 compared to the others, but this factor is not sufficient to discount the other 5 large positive eigenvalues.

**Table 2.** Building 1—Distance matrix of FTM distances and related eigenvalues.

Device	1	2	3	4	5	6	7	8	9	10	11	12	13	Eigenvalues
1	0	2409	4363	6108	1454	4022	5501	2847	3936	4870	3471	6411	7610	$8.4768 \times 10^7$
2	2409	0	2266	3713	1894	2387	3959	4800	2971	3809	4732	4848	5734	$4.1676 \times 10^7$
3	4363	2266	0	1896	3624	1535	1969	7619	3520	2739	6702	3376	4514	$4.3061 \times 10^6$
4	6308	3713	1896	0	5273	3114	1582	9680	4880	4101	8321	3672	3557	$3.969 \times 10^6$
5	1454	1894	3624	5273	0	2712	4211	2813	2786	3467	2596	5166	6476	$2.8132 \times 10^6$
6	4022	2387	1535	3114	2712	0	1775	4796	2582	1620	4636	2734	3587	$5.3881 \times 10^5$
7	5501	3959	1969	1582	4211	1775	0	7472	4030	2717	6293	2241	2390	$-2.037 \times 10^{-8}$
8	2847	4800	7619	9680	2813	4796	7472	0	2787	4218	1745	5481	6283	$-28,396$
9	3936	2971	3520	4880	2786	2582	4030	2787	0	2007	1880	2940	4246	$-1.642 \times 10^5$
10	4870	3809	2739	4101	3467	1620	2717	4218	2007	0	3345	1277	2649	$-1.1989 \times 10^6$
11	3471	4732	6702	8321	2596	4636	6293	1745	1880	3345	0	3842	5296	$2.3985 \times 10^6$
12	6411	4848	3376	3672	5166	2734	2241	5481	2940	1277	3842	0	1464	$-3.4372 \times 10^6$
13	7610	5734	4514	3557	6476	3587	2390	6283	4246	2649	5296	1464	0	$-2.1794 \times 10^7$

The measured FTM matrix generated from the other buildings present the same type of difficulty. Building 2 (also office building, but with 30 APs), the matrix surfaces 9 large eigenvalues. In building 3 (restaurant with partial high ceiling), 6 large positive eigenvalues are seen, obfuscating the fact that the deployment is three-dimensional. In building 4 (partial high ceiling warehouse), 11 large eigenvalues are seen. In building 5, 3 large eigenvalues are seen (while all the APs are on the same plane). This last observation may be caused by the strong obstacle in the center of the building, which reduces the number of measurable distances (although 14 APs are deployed, no AP/FTM device has measurable distance to more than 8 other APs (FTM devices); i.e., throughout the matrix each AP has no measured distance to 8 to 11 APs). As such, it is clear that the distance matrix alone is not sufficient to assert the dimensionality of the space. However, complementing with RSSI evaluation easily solves the issue. In building 1, another AP (AP15) is positioned on the upper floor, above AP05, and another (AP16) above AP06. As the ISTAs and RSTAs exchange at constant power (20 dBm), and although the RSSI is an inaccurate representation of the received power, a simple linearization and comparison between the distance computed by FTM and the expected signal (in free path) at that distance shows that AP15 signal to any AP systematically appears 12 to 28 dB below the pairwise signal between these APs at equivalent distance. By contrast, AP16 to AP15 signal appears 10 to 25 dB above the signal from AP16 to AP01 or AP07. This result immediately indicates that AP15 and AP16 are on different floor from the others (and that both AP15 and AP16 are on the same floor). The same logic is applicable to the other buildings.

## 6.2. Geometric Wall Remover Phase

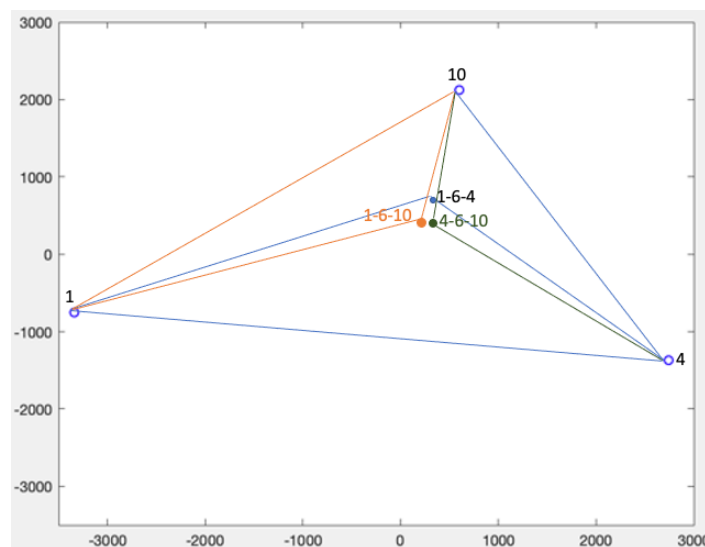
Table 3 displays the ratio between the mean of the measured distance and the ground truth in building 1. Highlighted in yellow, orange and red the distances that the geometric wall remover engine identifies as deviating from the others (small, medium or large

identified deviation, respectively). Some devices are out of range from one another (e.g., device 1 against sensor 13) and are not addressed in this phase.

**Table 3.** Measured distances to ground truth ratios, and pairs identified by the geometric wall remover method as “abnormal” in building 1.

Node	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.00	1.15	1.04	1.08	1.02	1.10	1.05	1.16	1.20	1.11	1.05	NaN	NaN
2	1.15	0.00	1.06	1.04	1.11	1.19	1.16	1.29	1.03	1.20	1.27	1.14	1.11
3	1.04	1.06	0.00	1.29	1.08	1.17	1.26	NaN	1.03	1.06	NaN	1.08	1.26
4	1.08	1.04	1.29	0.00	1.10	1.26	1.10	NaN	1.07	1.24	NaN	1.10	1.10
5	1.02	1.11	1.08	1.10	0.00	1.08	1.02	1.38	1.49	1.15	1.18	NaN	NaN
6	1.10	1.19	1.17	1.26	1.08	0.00	1.09	1.19	1.22	1.15	1.35	1.20	1.14
7	1.05	1.16	1.26	1.10	1.02	1.09	0.00	NaN	1.18	1.37	NaN	1.17	1.18
8	1.16	1.29	NaN	NaN	1.38	1.19	NaN	0.00	1.11	1.02	1.14	1.06	1.00
9	1.20	1.03	1.03	1.07	1.49	1.22	1.18	1.11	0.00	1.25	1.40	1.10	1.06
10	1.11	1.20	1.06	1.24	1.15	1.15	1.37	1.02	1.25	0.00	1.16	1.09	1.10
11	1.05	1.27	NaN	NaN	1.18	1.35	NaN	1.14	1.40	1.16	0.00	1.00	1.03
12	NaN	1.14	1.08	1.10	NaN	1.20	1.17	1.06	1.10	1.09	1.00	0.00	1.11
13	NaN	1.11	1.26	1.10	NaN	1.14	1.18	1.00	1.06	1.10	1.03	1.11	0.00

As can be seen, the geometric wall remover engine correctly identifies most incorrect distances, except those affecting devices positioned at the edge of the floor. A geometric illustration of this detection is represented in Figure 14, where stretches on AP6 position are detected through measuring its distance to APs 1, 4 and 10.



**Figure 14.** Stretches detected for AP6 position.

At the scale of the entire floor, the method initially identified pairs highlighted in red and orange in Table 3. A contraction factor  $\zeta$  was applied to each of these outliers (the step value of  $\zeta$  was chosen to be static and small, set to 0.98). As more iterations were run, each with an additional  $\zeta$  applied to identified distance dilations, the yellow pairs surfaced as abnormal. On the last iteration, the engine did not surface any more outlier, and the largest deviation from the ground truth (for the flagged pairs) was brought down to 1.13. This validation indicates that the geometric wall remover can correctly identify inner outliers and reduce the associated error, without distorting the graph (over or under reduction).

Comparable display is visible in the other buildings. FTM devices near APs on high ceiling display large distance stretches when ranging against devices near APs on lower ceilings, but devices at the edge of both domains display LoS distances and link both domains. The Geometric Wall Remover correctly identifies the high ceiling / low



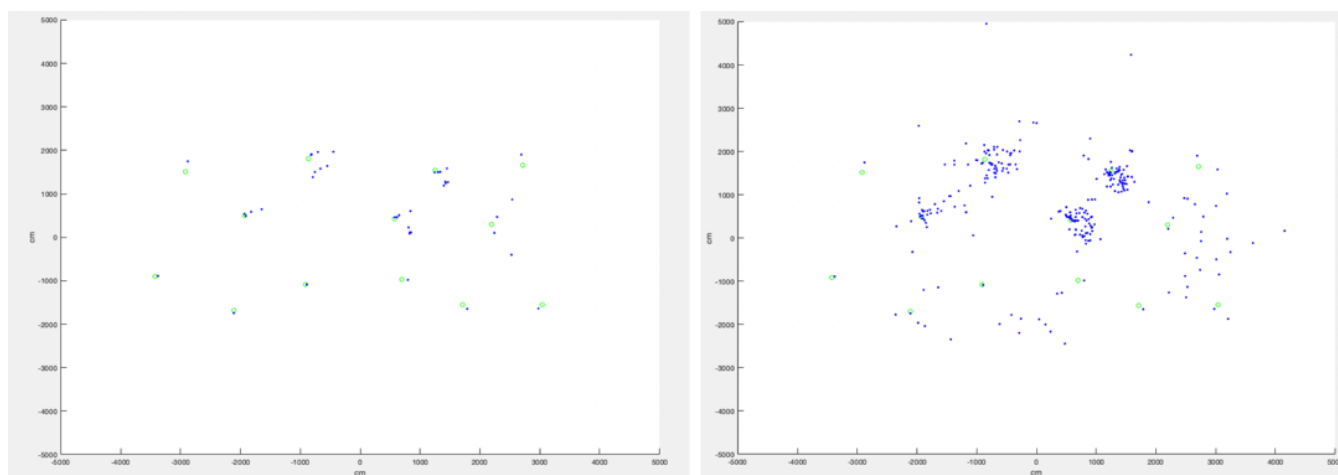
ceiling distant pairs as stretched, which a subset is illustrated in Figure 15. Here as well, highlighted in yellow, orange and red the distances that the geometric wall remover engine identifies as deviating from the others (small, medium or large identified deviation, respectively). High ceiling APs are 1 to 4, APs in low ceiling are 11 to 15, in italic.

Node	1	2	3	4	11	12	13	14	15
1	0.00	1.03	1.02	1.01	1.08	1.12	1.29	1.43	1.47
2	1.03	0.00	1.04	1.03	1.31	1.42	1.44	1.51	1.49
3	1.02	1.04	0.00	1.02	1.11	1.17	1.27	1.39	1.45
4	1.01	1.03	1.02	0.00	1.34	1.36	1.47	1.49	1.54
11	1.08	1.31	1.11	1.34	0.00	1.06	1.07	1.15	1.21
12	1.12	1.42	1.17	1.36	1.06	0.00	1.09	1.10	1.11
13	1.29	1.44	1.27	1.47	1.07	1.09	0.00	1.06	1.13
14	1.43	1.51	1.39	1.49	1.15	1.10	1.06	0.00	1.09
15	1.47	1.49	1.45	1.54	1.21	1.11	1.13	1.09	0.00

**Figure 15.** Measured distances to ground truth ratios, and pairs identified by the geometric wall remover method as ‘abnormal’ in building 3, lower half.

### 6.3. Iterative Algebro-Geometric Phase

In this phase, we start with the largest possible matrix. Building 1 has 13 APs and FTM devices, and with missing distance pairs (out-of-range APs), the system iteratively finds that a matrix of size 9 starts allowing for multiple usable solutions to appear with different individual pivots, as displayed in Figure 16 (left).

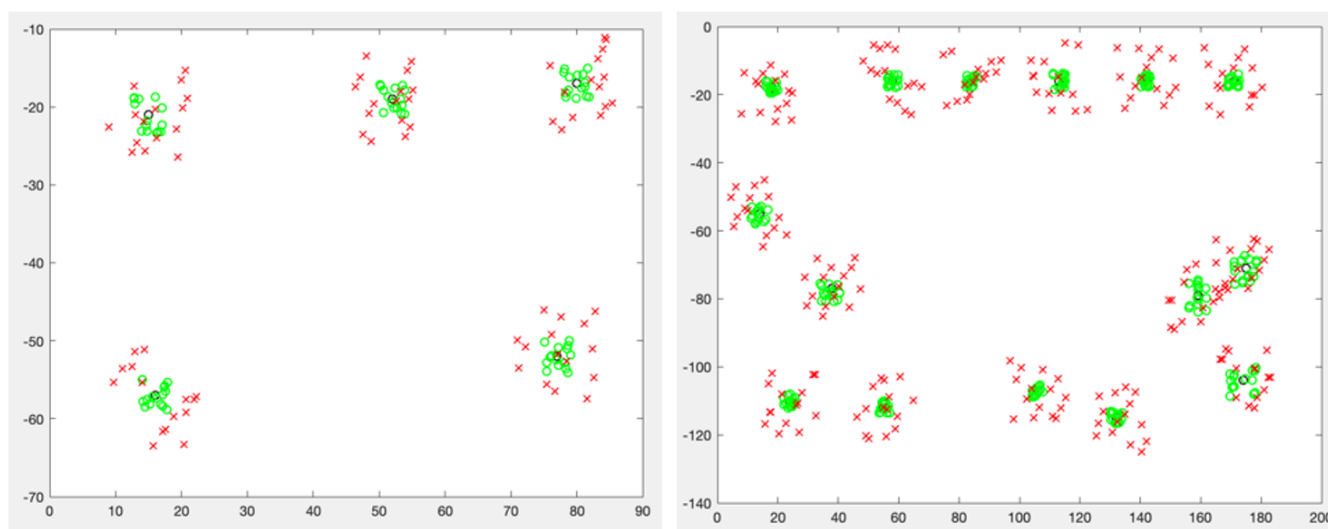


**Figure 16.** size 9 matrix positions (left), and size 8 matrix positions (right), before outlier filtering, in building 1.

Some points are used as pivots for all combinations in this iteration (and thus display a single position). As the matrix size decreases, more combinations appear. As the pivots change, noise also appears, as is visible in Figure 16 right, displaying the output of a matrix of size 8.

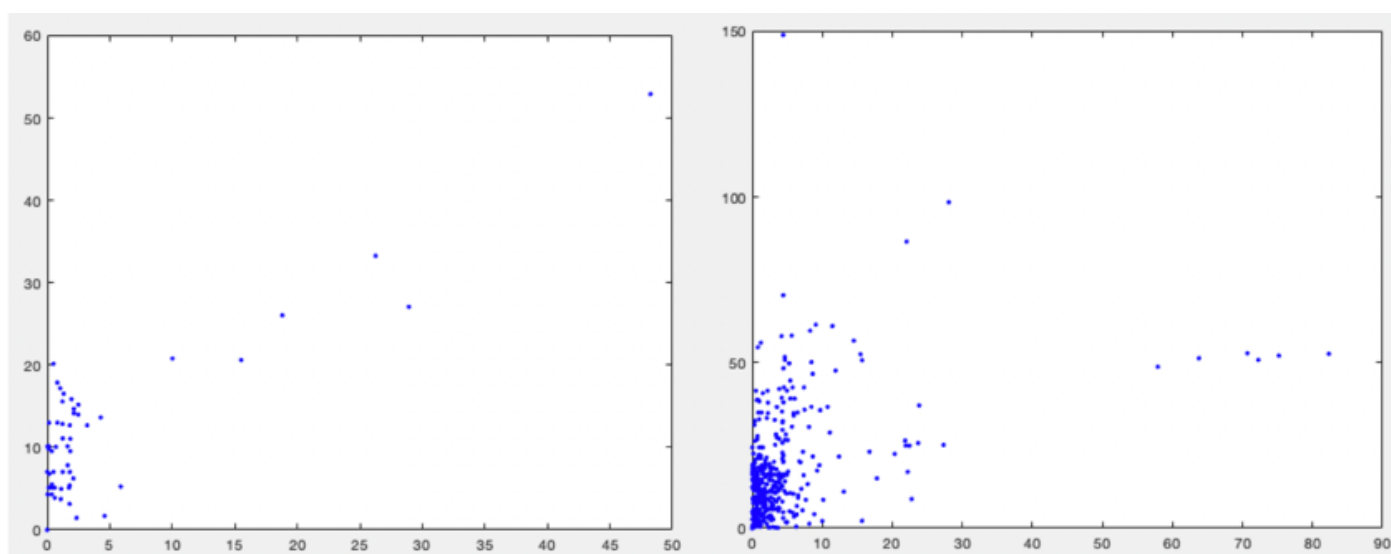
The process repeats iteratively. With smaller matrices and more combinations, clusters start to appear for each sensor, with various densities. In buildings 3 and 4, APs and devices in individual zones (low ceiling or high ceiling) are in LoS condition to each other. This is visible in Figure 17 (left), where a subset of building 3 is displayed (APs in the upper left part of the building). Green circles represent LoS (same height) matrices, while red crosses represent matrices with cross-height distances. The same density difference can be seen in building 5 (Figure 17 right), where APs/devices on the same side of the central block display close measurements (narrow clusters) while measurements through the block, when they succeed, display large variations (larger clusters).





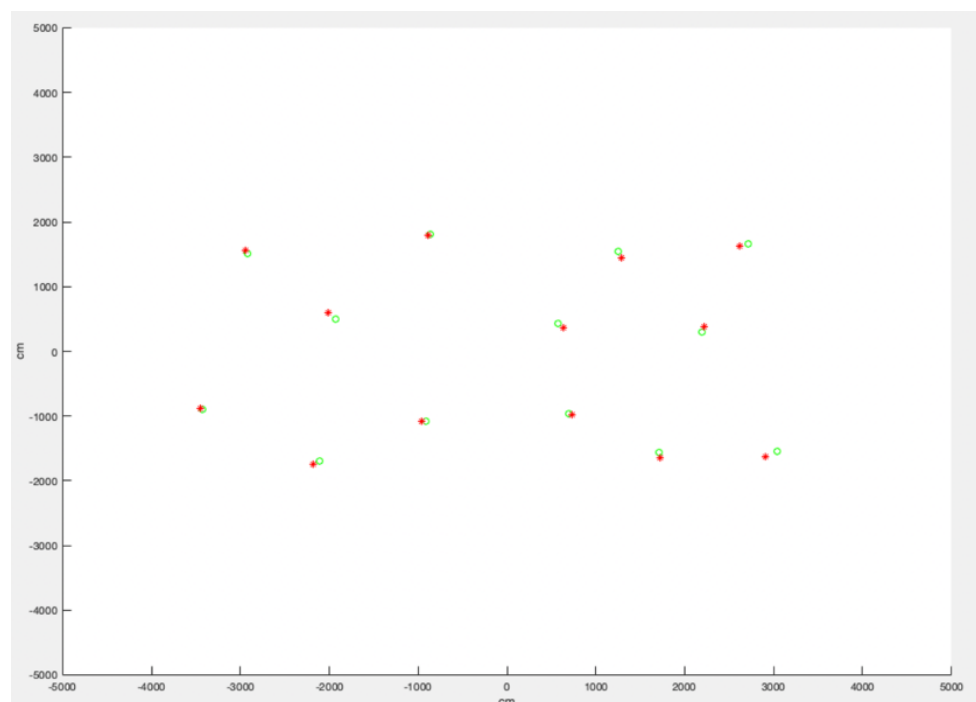
**Figure 17.** Bldg. 3 (left, partial) and Bldg. 5: position graph resulting from matrices built with APs in LoS (green circles) or nLoS (red crosses).

Therefore, for any cumulated graph, cluster centers can be defined and the mean cluster radius  $r_i$  compared between clusters. Then, outliers (points more than  $2\sigma$  away from the cluster center) can be removed from each cluster, not only within each cluster individually, but also for individual pairs, where deviation from the mean pair distance can be used to identify matrix combinations that surface large individual error. As a matrix of size  $m$  is used to compute positions, the position for each device is compared through each iteration, and the difference between the coordinates found at each iteration can be graphed, as shown in Figure 18 for building 1. As the matrix size gets smaller, the noise increases, and so do deviations between computed positions, as can be seen on the right side of Figure 18. We found that starting from matrices as large as possible was computationally more efficient (as there are less large matrix combinations than small matrix combinations), and that accurate location was found with matrix of 6 or more for building 1. Smaller matrices only marginally improved the results, and the computation cost increased. The same logic was observed for the other buildings, where matrices containing at least half the floor devices (or more) produced better results.



**Figure 18.** device 9 position deviations with matrix of 6 (left) and matrix of 5 (right).

After removing the outliers, each cluster center is re-computed. The final cluster centers are displayed in Figure 19. Ground truth positions are green circles, the computed cluster position for each device is represented as a red star. The maximum error is observed at 1.1143 meter in building 1, 1.2194 meter in building 2, and 0.9621 meter in building 5. In building 3 and 4, error is small in areas at the same height (0.7304 meter and 0.6812 meters respectively). When treating the APs / devices at different height as different floor levels, this result can be sufficient to move on to the next phase.



**Figure 19.** cluster center position computation (red star) vs ground truth position (green circles) in Building 1.

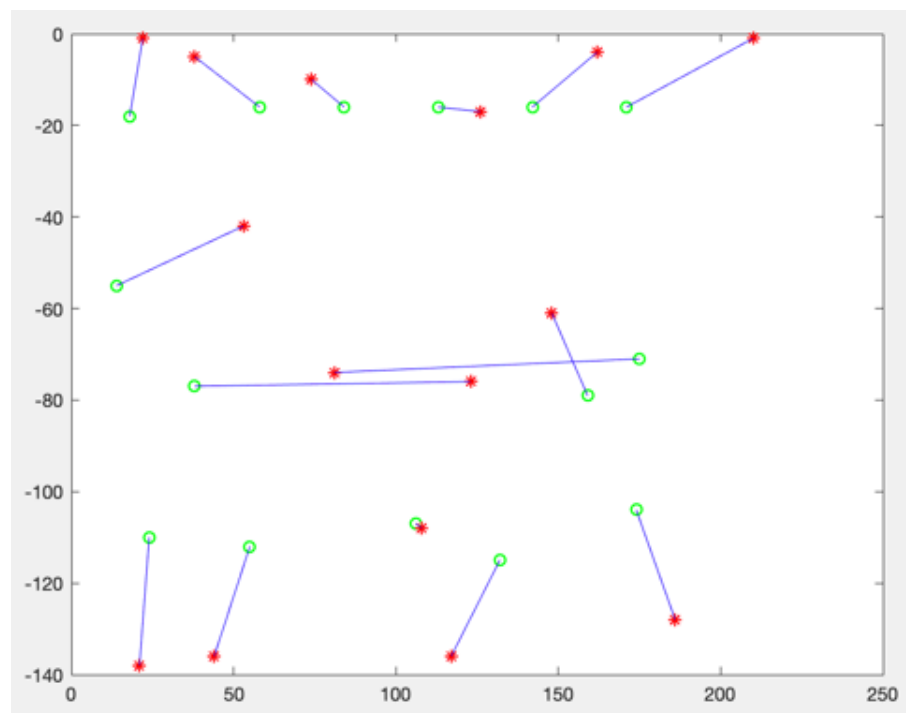
At this stage, the relative device positions are estimated, but their orientation is not known. However, using the ranging information from one or more mobile phones, walking outside the building, to two or more FTM devices, and sending to the device the GPS location of the mobile phone as seed, the graph can be rotated to its correct orientation and the phone GPS location can be used to populate the LCI values of all devices on the graph (and therefore all APs).

#### 6.4. Comparison with Other Methods

Noisy EDM completion is a complex problem that has been explored in multiple contexts. We will limit our comparison to the main methods listed in [21]. Naturally, using classic MDS boils down to performing an eigenvalue decomposition and geometric centering. As the matrix is noisy, the error is large. Additionally, the algorithm interprets missing distances as 0 values, which introduces irreconcilable inconsistencies in the matrix in all dimensions. The effect is worsened when the projection is constrained into  $\mathbb{R}^2$ . This affect can be attenuated by constructing and overlapping partial matrices for which distances are known [22]. Even in this case, the effect of asymmetry can be dire.

Many optimizations methods aim at finding missing elements from a matrix, but that is assumed to be noiseless. This assumption is necessary to maintain convexity, with the consequence of being unusable in our scenario. For incomplete and noisy matrices, a classical solution is to use semidefinite relaxation. This technique has multiple variants to adapt to different matrix sizes or sparsity scenarios, and both [22] and [23] are typical illustration of the associated reasoning. In all cases, the goal is to bound the rank of the Gram matrix to the target dimension space, thus constraining the number of positive and

non-null eigenvalues. The process is efficient, especially for large matrices. As it proceeds iteratively, it also has the virtue of minimizing the error. In buildings where measurements errors are averaged over a large number of APs/FTM devices, the error may also be simplified to Gaussian. However, FTM works with a limited number of devices, making the limitation obvious. When the stretch is asymmetric, the error reaches a point where the method is not usable anymore. Building 5 is a typical example, where a strong obstacle is in the middle of the block, while light walls do not prevent the signal from reaching the neighboring classes. The result of classical MDS projection for this building is visible in Figure 20.



**Figure 20.** classical MDS projection (after rotation) in building 5.

## 7. Conclusions

We presented a method to solve noisy Euclidian distance matrices (EDM). We showed that in the case of sensor time-of-flight measurements, measured distances are dilations of ground truth distance, but that the dilations are asymmetric, thus rendering classical EDM methods generally inaccurate, as they tend to assume noisy symmetry and therefore tend to center the error. We propose a machine learning method for identifying and reducing noise asymmetry based on evaluation of angles within overlapping triangles. We then propose a second machine learning method, aimed at graphing the positions of sensors derived from distance sub-matrices, and at identifying and removing combinations that surface excessive distances, thus progressively removing edge asymmetries and reducing the distance errors. We show that this method outperforms standard EDM resolution methods.

**Author Contributions:** Conceptualization, J.H., I.H.; methodology, J.H., I.H.; software, J.H., I.H.; validation, J.H., R.L., Y.B.; formal analysis, J.H., R.L.; investigation, J.H.; data curation, I.H.; writing—original draft preparation, J.H.; writing—review and editing, Y.B., R.L., N.M.; visualization, J.H., I.H.; supervision, N.M.; project administration, N.M.; funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Authors can confirm that all relevant data are included in the article.

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

## References

1. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. 2016. Available online: [https://standards.ieee.org/standard/802\\_11-2016.html](https://standards.ieee.org/standard/802_11-2016.html) (accessed on 11 March 2021).
2. Yue, Y.; Ruizhi, C.; Liang, C.; Guangyi, G.; Feng, Y. A Robust Dead Reckoning Algorithm Based on Wi-Fi FTM and Multiple Sensors. *Remote Sens.* **2019**, *11*, 504. [CrossRef]
3. Dvorecki, N.; Bar-Shalom, O.; Banin, L.; Amizur, Y. A Machine Learning Approach for Wi-Fi RTT Ranging. In Proceedings of the 2019 International Technical Meeting of The Institute of Navigation, Reston, VA, USA, 28–31 January 2019; pp. 435–444.
4. Polk, Y.; Linser, M.; Thomson, M.; Aboba, B. Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information. Available online: <https://tools.ietf.org/html/rfc6225> (accessed on 11 March 2021).
5. Henry, J.; Montavont, N.; Busnel, Y.; Ludinard, R.; Hrasko, I. Sensor Self-location with FTM Measurements. In Proceedings of the 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Thessaloniki, Greece, 12–14 October 2020; pp. 1–6. [CrossRef]
6. Lee, L.; Jones, M.; Ridenour, G.S.; Bennett, S.J.; Majors, A.C.; Melito, B.L.; Wilson, M.J. Comparison of Accuracy and Precision of GPS-Enabled Mobile Devices. In Proceedings of the 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, Fiji, 8–10 December 2016; pp. 73–82.
7. Liu, X.; Nath, S.K.; Govindan, R. Gnome: A Practical Approach to NLOS Mitigation for GPS Positioning in Smartphones. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, Munich, Germany, 10–15 June 2018.
8. Ma, L.; Zhang, C.; Wang, Y.; Peng, G.; Chen, C.; Zhao, J.; Wang, J. Estimating Urban Road GPS Environment Friendliness with Bus Trajectories: A City-Scale Approach. *Sensors* **2020**, *20*, 1580. [CrossRef] [PubMed]
9. Mun, S.; An, J.; Lee, J. Robust Positioning Algorithm for a Yard Transporter Using GPS Signals with a Modified FDI and HDOP. *Int. J. Precis. Eng. Manuf.* **2018**, *19*, 1107–1113. [CrossRef]
10. Zhao, C.; Wang, B. A UWB/Bluetooth Fusion Algorithm for Indoor Localization. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 4142–4146. [CrossRef]
11. Wang, P.; Luo, Y. Research on WiFi Indoor Location Algorithm Based on RSSI Ranging. In Proceedings of the 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 21–23 July 2017; pp. 1694–1698. [CrossRef]
12. Dattorro, J. *Convex Optimization Euclidean Distance Geometry 2e*; Meboo Publishing: Palo Alto, CA, USA, 2019; ISBN 9780578161402.
13. Zhang, K.; Shen, C.; Zhou, Q. A combined GPS UWB and MARG locationing algorithm for indoor and outdoor mixed scenario. *Clust. Comput.* **2019**, *22*, 5965–5974. [CrossRef]
14. Kjærgaard, M.B.; Blunck, H.; Godsk, T.; Toftkjær, T.; Christensen, D.L.; Grønbaek, K. Indoor Positioning Using GPS Revisited. *Int. Conf. Pervasive Comput.* **2010**, *2010*, 38–56.
15. Liberti, L. Distance Geometry and Data Science. *TOP* **2020**, *28*, 271–339.
16. Eren, T. Rigidity, computation, and randomization in network localization. *IEEE Infocom.* **2004**, *4*, 2673–2684.
17. Doherty, L.; Pister, K.S.J.; Ghaoui, L.E. Convex position estimation in wireless sensor networks. In Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Society, Anchorage, AK, USA, 22–26 April 2001; pp. 1655–1663.
18. Jacobs, D. Multidimensional Scaling: More Complete Proof and Some Insights Not Mentioned in Class. 2016. Available online: <http://www.cs.umd.edu/~djacobs/CMSC828/MDSexplain.pdf> (accessed on 11 March 2021).
19. de Leeuw, J. Applications of Convex Analysis to Multidimensional Scaling. 2005. Available online: <https://escholarship.org/uc/item/7wg0k7xq> (accessed on 11 March 2021).
20. Tsang, J.; Pereira, R. Taking All Positive Eigenvectors Is Suboptimal in Classical Multidimensional Scaling. *arXiv* **2016**, arXiv:abs/1402.2703.
21. Dokmanic, I.; Parhizkar, R.; Ranieri, J.; Vetterli, M. Euclidean Distance Matrices: Essential theory, algorithms, and applications. *IEEE Signal Process. Mag.* **2015**, *32*, 12–30. [CrossRef]
22. Candès, E.; Plan, Y. Matrix Completion With Noise. *Proc. IEEE* **2010**, *98*, 925–936. [CrossRef]
23. Guo, X.; Chu, L.; Sun, X. Accurate Localization of Multiple Sources Using Semidefinite Programming Based on Incomplete Range Matrix. *IEEE Sens. J.* **2016**, *16*, 5319–5324. [CrossRef]