Large Network UWB Localization: Algorithms and Implementation

Nakul Garg[1], *Irtaza Shahid*[1], *Ramanujan K Sheshadri*[2], *Karthikeyan Sundaresan*[3], *Nirupam Roy*[1]
[1]*University of Maryland, College Park,* [2]*Nokia Bell Labs,* [3]*Georgia Institute of Technology*

Abstract

Localization of networked nodes is an essential problem in emerging applications, including first-responder navigation, automated manufacturing lines, vehicular and drone navigation, asset tracking, Internet of Things, and 5G communication networks. In this paper, we present *Locate3D*, a novel system for peer-to-peer node localization and orientation estimation in large networks. Unlike traditional range-only methods, *Locate3D* introduces angle-of-arrival (AoA) data as an added network topology constraint. The system solves three key challenges: it uses angles to reduce the number of measurements required by 4× and jointly uses range and angle data for location estimation. We develop a spanning-tree approach for fast location updates, and to ensure the output graphs are rigid and uniquely realizable, even in occluded or weakly connected areas. *Locate3D* cuts down latency by up to 75% without compromising accuracy, surpassing standard range-only solutions. It has a 0.86 meter median localization error for building-scale multi-floor networks (32 nodes, 0 anchors) and 12.09 meters for large-scale networks (100,000 nodes, 15 anchors).

1 Introduction

A swarm of connected nodes is the underlying architecture for many emerging applications. Miniaturized sensing modules are scattered like seeds [37, 79] or carried by insects [7, 8, 38] to scale a vast region for fine-grained sensor networks toward border protection [10], animal migration tracking [44], or precision agriculture [23, 24, 81]. Flocks of drones can localize each other to fly in formation for charting inaccessible regions [22, 34, 71] or for airshows [2]. A smart network of tags can enable city-scale tracking of deliveries or missing objects in real-time [52]. Future of the cellular [21, 68] and vehicular networks [4] can pave the path toward autonomy [86, 95] and road safety [90]. Localization of the nodes in such large networks is an essential requirement and at the same time a challenging problem when the number of nodes is large. In this paper, we present *Locate3D*, a system that is grounded on both theoretical and practical foundations, providing a reliable framework for fast peer-to-peer localization as well as orientation of nodes on a large network.

Multidimensional scaling (MDS) has been the essence of nearly all large-scale localization algorithms for wireless networks. The primary reason MDS is adapted so widely is its ability to reconstruct the relative map of the network nodes with little infrastructural support, even in the form of the anchor nodes with known locations. MDS-based algorithms are fundamentally centralized ranging-based systems that consider inter-node distances in the Euclidean space to optimize for sensor locations. It, however, suited the capacity of the mobile nodes that can use the time of flight of the signal or RSSI-based model to estimate ranges without additional hardware requirements or computational complexity. However, recent mobile nodes have evolved not only to accurately sense the ranges but are also equipped with multi-dimensional antenna arrays to estimate reliable angles of the nodes. For instance, around 20 years after being released for commercial applications in 2002 [19], off-the-shelf UWB sensors can now sense the angle of the received signal [65, 66]. Unfortunately, the entire class of the MDS-based network localization algorithms can not take advantage of this new-found capability of the mobile nodes. Intuitively the angle information of the peer nodes can serve as additional constraints of the network topology leading to faster convergence to the location estimation. Moreover, the range and angles are estimated simultaneously per exchange of signals between nodes without incurring additional measurement latency. This convenient information is wasted as the MDS objective function can not jointly optimize on the Euclidean distance and angle plane. We develop a new network localization algorithm with a redefined objective function to include range and angle together for optimization to bridge this gap.

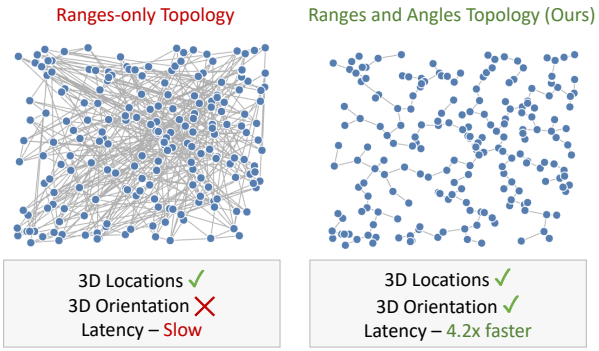


Figure 1: *Locate3D*'s approach to include both angle and edge constraints for faster and efficient localization.

The scalability of a network localization solution depends on several practical factors beyond the correctness of the theoretical formulations. The level of dependency on the infrastructure is the most crucial of them. In addition, a dynamic topology of mobile nodes requires short update latency of location estimations. Like any peer-to-peer localization system, *Locate3D* requires at least four anchor nodes for the unambiguous global 3D location of the nodes. However, the relative locations of the entire topology are correct with no assumption on the anchors. We enable our algorithms to incorporate any number of available anchor nodes and other infrastructural opportunities to create virtual anchors that enhance overall localization accuracy. In the proposed system, the joint range and angle-based optimization reduce the measurement and initial topology estimation latency, then a spanning-tree-based optimal edge selection expedites updates on locations after initial estimation, and finally, a graph rigidity-based solution makes the estimation robust to local occlusions or poor peer-to-peer connectivity.

A class of solutions for network localization resort to multimodal data to improve accuracy and reduce the update latency of the system. While effective in a small number of nodes and within restricted environmental conditions, the multimodality assumption limits the scalability of the system. It is infeasible to maintain homogeneous data quality with thousands of nodes spread across a large geographical region. For instance, some recent papers [51] use Visual Inertial Odometry (VIO) - a camera-based solution to track orientation - to improve localization accuracy. We argue that an unimodal solution is ideal for large networks in terms of consistency and ease of practical deployment and maintenance. It makes our solution equally applicable to the networks of resource-constrained low-power nodes. The core localization algorithm is, however, applicable to any modality that can measure the peer-to-peer range and relative angles, and naturally, the location accuracy is defined by the accuracy of the measurements. For instantiating the algorithm in a prototype and realistic large-scale simulations, we consider off-the-shelf Ultra-WideBand (UWB) wireless sensing nodes.

This paper strives to improve the accuracy of 3D localization of UWB-enabled nodes on a large single-modality network as shown in Figure 1. To this end, we have made the following three specific contributions at the current stage of this project:

- Developed a novel 3D network localization algorithm that jointly incorporates range and angle information for topology estimation. This algorithm achieves a 75% latency improvement and a median localization accuracy of 12.09 meters for a 100,000-node network spanning several kilometers, using only 15 static anchors. In a 32-node, three-floor building-scale experiment, the algorithm demonstrates a localization accuracy of 0.86 meters.
- Designed a system with a supporting algorithm that addresses challenges of occlusions and limited sensor field of view. The system includes an optimal spanning tree estimation algorithm and a decomposition technique that segments non-rigid graphs into smaller rigid subgraphs, considering both range and angle constraints.
- Implemented a working prototype with UWB-enabled nodes and evaluated its performance in real-world scenarios, including room-scale and multi-floor building experiments. Demonstrated the system's scalability through a city-scale simulation using real-world UWB measurement traces.

2 System Design

Our system, *Locate3D*, uses pair-wise Ultra-Wideband (UWB) RF measurements, specifically range and angle-of-arrival (AoA) information, to determine the precise 3D positions of interconnected nodes in relation to one another. The system is designed using four key components:

Joint range-angle localization: We develop an optimization method that uses range and angle measurements as constraints to build a network of localized nodes. Incorporating angles ensures an accurate location estimation with fewer number of edges compared to a range-only method, effectively reducing the latency up to 4.2×.
Reference frame transformation: Each node's angle measurement is a composite value of both the relative orientation (determined by the antenna angle with respect to the peer's antenna) and the AoA. *Locate3D* efficiently decomposes this combined measurement to transform all node axes to a common frame of reference.
Scaling to large networks: We develop an edge selection strategy that ensures the formulation of the optimal spanning tree, facilitating the interconnection of large topologies. Large networks often contain flexible or unconstrained edges, which can result in isolated but structurally rigid subgraphs. We aim to identify and resolve these subgraphs separately.
Opportunistic integration with infrastructure: *Locate3D* can leverage any pre-existing infrastructure, incorporating it as anchor points to refine localization accuracy. We further extend our system's capability by introducing the concept of virtual anchors. This mechanism temporarily designates specific mobile nodes as anchors and adjusts edge weights, disseminating the accuracy throughout the network. Below, we discuss the individual algorithmic contributions in detail.

2.1 Joint Range-Angle Localization

We formulate the localization problem as an optimization problem that aims to minimize the difference between measured pairwise distances and the distances corresponding to the estimated coordinates. There is a plethora of work done in network localization using range measurements only.

SMACOF [15, 33], non-metric MDS etc [9, 61, 78]. Some recent works also account for missing edges [72, 73] and NLOS cases [14, 17, 82]. However, none of the works have jointly incorporated AoA information in the network.

Problem formulation: Consider a topology of N nodes (mobile devices), with unknown locations $X = [(x_1, y_1, z_1), \cdots (x_N, y_N, z_N)]$. Suppose the nodes measure the range between each other, $\hat{r_{ij}}$ where (i, j) is the pair nodes i and j in the topology. We aim to solve for X while minimizing the cost function:

$$\min_X \sum_{i,j} w_{ij}(\hat{r_{ij}} - r_{ij}(X))^2 \tag{1}$$

where, w_{ij} is the weight assigned to edge between nodes i and j and $r_{ij}(X)$ is the euclidean distance between them given by Equation 2. We set $w_{ij} = 0$ for missing edges.

$$r_{ij}(X) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{2}$$

Adding angles in the topology: Incorporating angles into the network topology offers the benefit of reducing the necessary number of edges, which subsequently decreases latency. This approach, however, is not as straightforward due to the highly non-convex and discontinuous nature of angular loss functions when simply added with the range-based loss function.
Existing works [11,50,64] which use L1 and L2 losses directly on this loss function show that the objective function has many local minima.
We can compute the angular loss using the function $f(X)$, expressed as:

$$f(X) = \left[\hat{\theta_{ij}} - \arctan\left[\frac{y_i - y_j}{x_i - x_j}\right]\right]^2 \tag{3}$$

This function encapsulates the difference between the measured angles, denoted by $\hat{\theta_{ij}}$, and the angles calculated from the estimated coordinates. The non-convex nature of $f(X)$ occurs due to the arctangent and the least square operation on angles. This results in multiple local minimums, making it highly prone to generating inaccurate topologies. The primary contributor to this issue is the restrictive interval $[-\pi/2, \pi/2]$ that the arctangent function operates in, failing to account for points in the left quadrants of the plane, thereby resulting in the same angles for coordinates (x, y) and $(-x, -y)$. To address this, we first use the 2-argument arctangent, a variant of the arctangent function that considers both x and y inputs and adjusts for the signs, hence returning angles within the interval $[-\pi, \pi]$.
Unfortunately, the transformation is not enough, as the gradient consists of non-differential making it prone to getting ensnared in local minimums. To overcome this, we apply another transformation to the loss function. We take the negative cosine of the angles, creating a smoother, continuous, and differentiable function restricted within the $[0, 1]$ range.

This transformation of the new loss function, is defined as:

$$f(X) = 1 - cos(\hat{\theta_{ij}} - arctan2(y_i - y_j, x_i - x_j)) \tag{4}$$

Finally, we combine the range and angular loss functions to efficiently integrate angles in the network topology optimization formulating a joint optimization problem.

$$\min_X \left[\frac{\sum_{i,j} w^r_{ij}(\hat{r_{ij}} - r_{ij}(X))^2}{\sum_{i,j} \hat{r_{ij}}} + \sum_{i,j} w^{\theta}_{ij}(1 - cos(\hat{\theta_{ij}} - \theta_{ij}(X)))\right] \tag{5}$$

where, $\theta_{ij}(X) = arctan2(y_i - y_j, x_i - x_j)$. Note that we scale the range loss function with $(\sum_i \sum_j \hat{r_{ij}})$. The range loss can be much higher compared to the angular loss, which is in the range $[0, 1]$, and can dominate the overall gradient/overshadow the angular loss. Hence, we normalize the range loss function before adding the angular loss part.

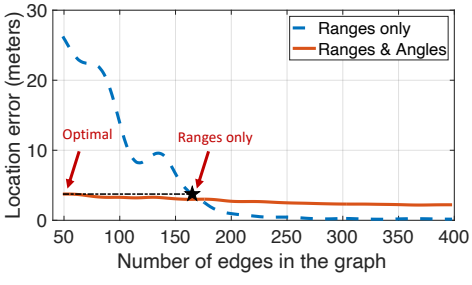


Figure 2: Comparative analysis of constraints: Incorporating angles reduces the number of edges required to attain the same level of accuracy as the "Ranges only" approach.

We conducted a simulation to evaluate the effectiveness of integrating angular measurements with distance constraints in topology optimization. The simulation involved a 50-node network within a $100m \times 100m$ area, incrementally adding edges under two constraint scenarios: ranges only, and both ranges and angles. Figure 2 shows that using only ranges required more than $3\times$ the number of edges to achieve accuracy comparable to our method combining ranges and angles. However, as more edges were added, the range-only approach eventually surpassed the range-angle method in accuracy. This can be attributed to the use of raw angles, which may include noisy measurements, and the random selection of edges rather than optimal spanning tree edges. Despite this, the method promises latency improvement over range-only methods without compromising accuracy even for a sub-optimal set of edges. We address the challenges related to noisy angle measurements and edge selection in Section 2.2, where we present techniques for filtering erroneous angles and constructing optimal spanning trees.

2.2 Scaling to Large networks

In this section, we detail how *Locate3D* can localize large node networks efficiently. To adapt to city-scale networks, we address three significant challenges linked with large networks.

• *Optimal edge selection*: Identifying the optimal edges to sample in the graph topology is vital to forming a unique, rigid structure while optimizing latency and accuracy.

• *Handling erroneous angles*: While in theory, all measured edges should have angle data, the limited FOV of commercial antenna arrays can sometimes result in inaccurate angle information. Correcting these errors is vital as they can adversely affect the graph's rigidity constraints.

• *Rigidity decomposition*: Large networks frequently consist of flexible or unconstrained edges which leads to smaller but rigid subgraphs. We need to identify and solve these subgraphs individually.

Next, we will elaborate on how *Locate3D* mitigates these challenges to scale to larger networks, introducing techniques that could reduce *Locate3D*'s latency by up to 4.2×.

2.2.1 Optimal Edge Selection

During any measurement time iteration, *Locate3D* compiles a list of optimal edges that are to be measured within the graph. These edges are chosen based on three criteria - (1) They form a rigid and unique topology, eliminating flexibility or ambiguity. (2) They are minimal in number to decrease overall latency. (3) They are feasible, meaning the edges fall within the necessary range and angle FoV.

In a topology comprising n nodes, a maximum of $n(n-1)$ edges are potentially available. However, acquiring data for each possible edge isn't viable as it is time-consuming and increases exponentially as n increases. Hence, rather than overconstraining the topology, we can create the topology with significantly fewer edges. According to Laman's condition [42], for a system that doesn't utilize angles, the minimum number of edges equals to $2n-3$ and $3n-4$ for 2D and 3D setups respectively. Interestingly, our approach necessitates only a *single* edge per node to constrain it, amounting to only $n-1$ edges in total. This efficiency stems from the fact that a single edge contains three constraints - range, azimuth angle, and elevation angle. Moreover, any random $n-1$ edges may not always be enough, as some edges could be redundant. Rather we need *well-distributed* $n-1$ edges to make the topology rigid and accurate. This renders our problem closely analogous to the *Minimum Spanning Tree problem* found in graph theory [28].

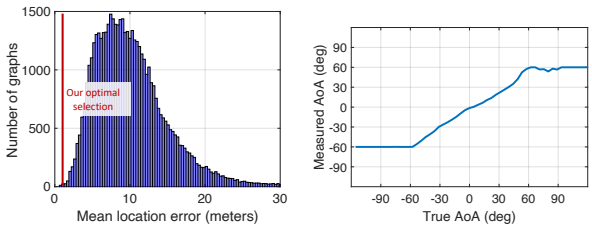
Minimum spanning tree: The minimum spanning tree (MST) is a subset of the graph's edges connecting all nodes with the least total edge weight. In our system, we utilize Kruskal's algorithm, a greedy algorithm that arranges the graph's edges in increasing order of their weights and adds edges to the MST as long as they do not form a cycle, thus guaranteeing the minimum combined edge weights.

Edge weight calculation: Our weight assignment algorithm minimizes localization error in the MST by utilizing prior knowledge of the network topology from the previous iteration, including estimated range (r) and angle (θ) values between nodes. The *area of uncertainty*, denoted as A, is calculated using sector area, $A = \pi(\alpha\sigma_r)^2\frac{\alpha\sigma_\theta}{2\pi}$, where σ_r and σ_θ are the standard deviations of the range and angle estimates, respectively, and α is a factor that scales the standard deviations based on LOS ($\alpha = 1$) or NLOS ($\alpha > 1$) conditions. The edge weight, w, is then computed as $w = A$, representing the potential location uncertainty based on the estimated range, angle, and their associated error distributions. Edges with shorter distances and smaller standard deviations inherently have a smaller area of uncertainty, and edges with available angle information are favored over those restricted to range data only. Figure 3a demonstrates that our algorithm's optimal spanning tree ranks within the top 1% of all possible spanning trees in terms of localization error. Algorithm 1 uses these estimated range and angle values to compute edge weights and form the MST, iteratively selecting the lowest-weight edges without introducing cycles. Actual range and angle measurements are taken for the selected edges once the MST is formed. During the cold start phase, when no prior knowledge is available, all nodes measure their neighbors, similar to traditional systems like [51, 94], resulting in increased latency for the first iteration. Subsequent iterations efficiently select edges for measurement by leveraging the prior knowledge of the network topology.

Algorithm 1 Optimal Edge Selection Algorithm

Input: $G(V,E)$, $\hat{r}$, $\hat{\theta}$, σ_r, σ_θ, α **Output:** $M(V_{MST}, E_{MST})$

procedure
  for each E in G **do**
    Retrieve $\hat{r}$, $\hat{\theta}$, σ_r, σ_θ, and α for current E
    Assign weight $w(E) = \pi(\alpha\sigma_r)^2\frac{\alpha\sigma_\theta}{2\pi}$
  end for
  $G_{sorted}(V,E)$ = Sort($G(V,E)$, $w(E)$)
  Initialize $M(V_{MST}, E_{MST})$ as empty
  for each V,E in G_{sorted} **do**
    if adding V,E to M does not form a cycle **then**
      Add V,E to M
    end if
  end for
  return $M(V_{MST}, E_{MST})$
end procedure



(a) (b)

Figure 3: (a) Histogram of localization errors for all spanning trees. (b) Reported angles by COTS UWB sensor.

2.2.2 Handling Erroneous Angles

In practical settings, nodes may not consistently measure each other's ranges or angles due to limited sensing range or angular FOV. This is due to the limited FOV of antenna arrays present in commonly used UWB modules like Decawave [65, 66] and NXP [54]. These modules typically restrict the AoA field of view (FOV) to between 90° and 120° to maintain a respectable accuracy in measured AoA, primarily because the AoA estimation algorithm, which depends on the phase difference of the incoming signal, can result in significant errors when the AoA approaches the broadside of the antenna. Figure 3b shows the reported angles by an off-the-shelf UWB sensor [54]. To address this, *Locate3D* employs two strategies for handling erroneous angles. First, we filter and flag angles as erroneous if they fall outside the range of $|FOV_{angle} - tolerance|$, effectively removing angles close to the broadside. Second, when available, *Locate3D* tracks the rotation of each node using its inertial sensors. Utilizing this rotation data, we can determine if any two nodes are within each other's FOV and flag the angles as valid or erroneous accordingly. In scenarios where some nodes lack inertial sensors or AoA estimation capability, *Locate3D* can still perform localization using the filtered angle data from the nodes equipped with these capabilities.

2.2.3 Rigidity Decomposition

While previous sections operate under the presumption of network topology rigidity, ensuring a unique realization of topology with given constraints, this isn't always the case. Notably, the optimal spanning tree derived earlier doesn't inherently ensure rigidity. Even fully connected graphs can remain non-rigid due to absent constraints, as illustrated in Figure 4 where spanning trees lack rigidity due to missing angle constraints. To address this, we develop a rigidity-based graph decomposition technique that segments the graph into smaller *rigid subgraphs*. These subgraphs can then be solved with refined edge weights ensuring that the topology is both rigid and uniquely realizable.

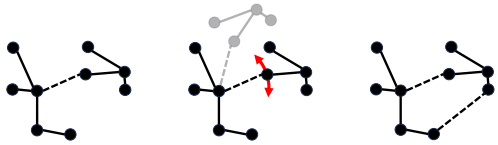


Figure 4: Different spanning trees representing rigid and non-rigid graphs. Solid lines indicate both range+angle edges, and dashed lines indicate range-only edges. (a) A connected but non-rigid graph due to missing angle information in an edge. (b) The subgraph is free to rotate. (c) Adding a range measurement makes the graph rigid.

Quantifying rigidity: The Rigidity of a topology determines whether it is possible to uniquely determine the location of all nodes in the topology with respect to other nodes. This leaves us with some trivial transformations of topology like translations and rotations in the space. The rigidity can be quantified using the degree of freedom (DoF) of the graph. Each node has three DoFs, representing movements in the x, y, and z directions. Thus, a graph with n nodes inherently has $3n$ DoFs. As constraints are imposed, the available DoFs decrease. A graph is deemed rigid if its DoFs are ≤ 3, with these residual DoFs accounting for the whole-graph translational motions.

Eigenanalysis of rigidity matrix: To identify independent and unconstrained motions that the topology can perform without violating the constraints, we perform the eigenvector analysis of the topology. To capture the constraints in a compact matrix form, we first introduce the rigidity matrix, R. Every row of R denotes a unique constraint equation, which could be a distance or angle constraint between nodes. Each column of R corresponds to the x, y and z coordinates for each node in the graph. Thus, for a graph having n nodes, the matrix expands to have $3n$ columns.
Each entry r_{ij} in this matrix is determined by the partial derivatives of the constraint equations 2 and 3, as given by:

$$r_{ij} = \begin{cases} \frac{\partial d_{ij}}{\partial x_m} & \text{if edge } ij \text{ is a distance constraint} \\ \frac{\partial \theta_{ij}}{\partial y_m} & \text{if edge } ij \text{ is an angle constraint} \\ 0 & \text{otherwise} \end{cases}$$

$$R = \begin{bmatrix} \frac{\partial d_{1,2}}{\partial x_1} & \frac{\partial d_{1,2}}{\partial y_1} & \cdots & \frac{\partial d_{1,2}}{\partial x_n} & \frac{\partial d_{1,2}}{\partial y_n} \\ \frac{\partial d_{2,3}}{\partial x_1} & \frac{\partial d_{2,3}}{\partial y_1} & \cdots & \frac{\partial d_{2,3}}{\partial x_n} & \frac{\partial d_{2,3}}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial \theta_{1,2}}{\partial x_1} & \frac{\partial \theta_{1,2}}{\partial y_1} & \cdots & \frac{\partial \theta_{1,2}}{\partial x_n} & \frac{\partial \theta_{1,2}}{\partial y_n} \\ \frac{\partial \theta_{2,3}}{\partial x_1} & \frac{\partial \theta_{2,3}}{\partial y_1} & \cdots & \frac{\partial \theta_{2,3}}{\partial x_n} & \frac{\partial \theta_{2,3}}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{bmatrix}$$

Let $\lambda_1, \lambda_2, \ldots, \lambda_m$ be the eigenvalues of the matrix $M = RR^T$, where M is the symmetric and positive semi-definite matrix of R. Topology is uniquely rigid if the number of zero eigenvalues of M is equal to $3n-3$. A number of zero eigenvalues, or the degree of freedom, can be represented as:

$$DoF = \sum_{i=1}^{m} \delta(\lambda_i), \quad \text{where } \delta(\lambda) = \begin{cases} 1 & \text{if } |\lambda| < \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Here, ε is a small positive number close to zero, chosen to account for numerical inaccuracies (e.g., due to floating-point representation limits in computers). If $|\lambda_i| < \varepsilon$, it is considered a zero eigenvalue. Figure 5 illustrates the eigenvectors corresponding to the three zero eigenvalues λ_1, λ_2, and λ_3 showing the motion of each node.

Constructing subgraphs: By examining the zero eigenvalues, we gain insights into the *displacement vectors* of each node. For a fully rigid topology, these vectors maintain consistent magnitudes and directions across nodes. In contrast, non-rigid graphs display varied displacements.

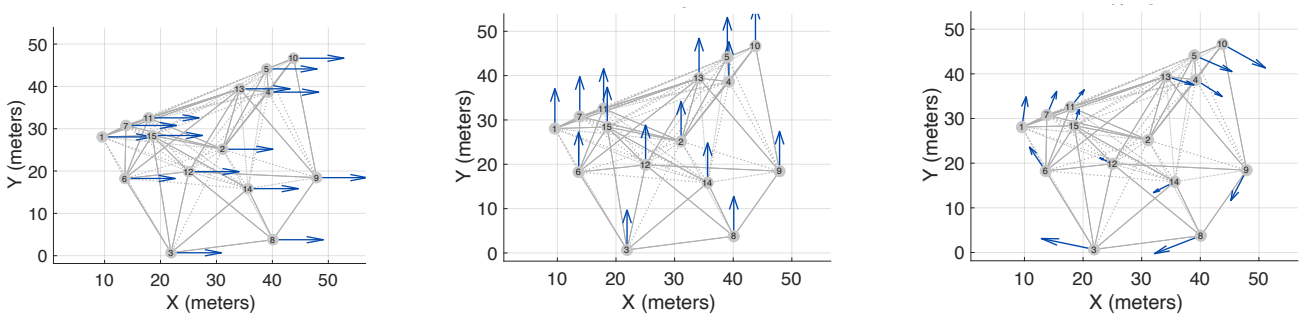


Figure 5: The displacements corresponding to zero eigenvalues represent the translational and rotational motions that the nodes can undergo without violating any constraint.

Our core intuition is that displacement vectors can be used for decomposition: nodes that have identical displacement vectors in terms of magnitude and direction inherently form a rigid subset. This means they can only undergo collective motion. To extract and group these subsets, we first reshape the eigenvector into a 3×n matrix to extract the displacement vectors of each node. Each row within this matrix signifies the displacement of a node across the three axes. Next, we identify the unique displacement vectors in this matrix where every unique vector signals the presence of a distinct "rigid subgraph" within the primary graph. By grouping the nodes corresponding to unique displacement vectors, we find the vertices belonging to the associated "rigid subgraph".

Critical edges list: Beyond recognizing rigid subgraphs, our approach also pinpoints the edges connecting the separate subgraphs. We term these edges as 'critical edges', as these edges are crucial for rigidity, serving as the connectors between independent rigid substructures. For enhanced analysis in subsequent iterations, we also prioritize these critical edges based on their latest computed locations. This prioritized list of edges underlines those edges that are more likely to come into proximity with one another, thus enhancing the overall scalability and rigidity of the graph.

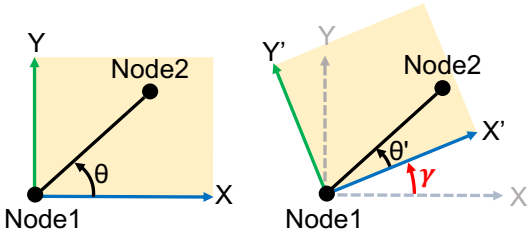


Figure 6: (a) When Node 1 is aligned with the global frame of reference, it reports that Node 2 is positioned at angle θ. (b) When Node 1 is rotated by γ relative to the global frame of reference in a 3D space, it reports a distinctly different angle θ′, for the same node.

2.3 Reference Frame Transformation

In the above section, we assume that all nodes share a common frame of reference. Here, we relax this assumption and address the challenges arising from differing local frames of reference for each node. Every node measures angles based on two factors: its relative orientation (defined by its antenna angle concerning another node's antenna) and the AoA. The challenge lies in the fact that these angle measurements are in the node's local frame of reference, which moves according to the node's orientation within a broader, global context. Consider the 2D example in Figure 6. In scenario (a), Node 1 identifies Node 2 at an angle θ. But in scenario (b), after Node 1 turns by an angle γ, it sees Node 2 at a different angle θ′. This change in reported angles is due to Node 1's rotation, even though the positions of the nodes didn't change. This issue gets complex in 3D, where nodes can rotate in any direction. Our goal is to deconstruct these angle measurements from orientation offsets and align all nodes to a singular, shared frame of reference.

2.3.1 Estimating Orientation Offsets

Every node's orientation in a 3D space is defined by a set of three rotation angles. Specifically, they depict the node's rotations about its X, Y, and Z axes and are recognized as the Roll(α), Pitch(β) and Yaw(γ) [89].

To estimate a node's orientation, we leverage a fundamental observation: When the orientations of nodes are in the same global frame, the measured azimuth AoA (θ) and elevation AoA (ϕ) are complementary, implying they obey the condition: $\theta_{ij} - \theta_{ji} = \pi$ and $\phi_{ij} = -\phi_{ji}$ (Figure 7). Using this insight, we formulate an optimization problem with the goal of estimating the rotation offsets (α, β, γ). The objective is to minimize the deviation from the above-mentioned constraints:

$$\min_{\alpha,\beta,\gamma} \sum_{i,j} (\hat{\theta}_{ij} - \hat{\theta}_{ji} - \pi) + (\hat{\phi}_{ij} + \hat{\phi}_{ji})$$

In this equation, $\hat{\theta}_{ij}$ and $\hat{\phi}_{ij}$ denote the rotated azimuth and elevation angles after the node has undergone rotation by α_i, β_i, and γ_i. Solving for this optimization provides us with the required orientation offsets to align all nodes in the system with the global frame of reference. The residual angles, $\hat{\theta}$ and $\hat{\phi}$, thus correspond to the AoAs in the global frame.

Next, we break down the steps in the optimization problem, detailing how we compute $\hat{\theta}$ and $\hat{\phi}$ through the rotation of local frames. The rotation in a 3D context can be decomposed into three unique rotations around the x, y, and z basis axes.

Step1 - Determine the unit vector based on the estimated

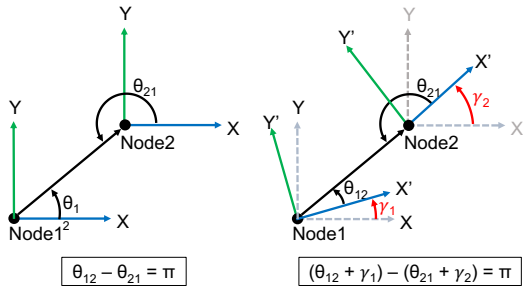


Figure 7: The relation between AoAs reported by two nodes (a) when nodes are perfectly aligned, and (b) when nodes are oriented with angle γ relative to the global frame of reference.

azimuth and elevation angles:

$$U = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} cos(\phi) * cos(\theta) \\ cos(\phi) * sin(\theta) \\ sin(\phi) \end{bmatrix}$$

Step2 - Apply the rotation matrix [76] to adjust the unit vector using the computed roll, pitch, and yaw angles (α, β, and γ). This results in a new unit vector, V, which reflects the node's orientation post-adjustment.

$$V = R_z(\gamma)R_y(\beta)R_x(\alpha)U$$

$$V = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & -\cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & -\cos\alpha \end{bmatrix} U$$

Step3 - Compute the azimuth and elevation angles from the rotated vector V to obtain the corrected AoA measurements.

$$\begin{bmatrix} \hat{\theta} \\ \hat{\phi} \end{bmatrix} = \begin{bmatrix} tan^{-1}(y/z) \\ tan^{-1}(z/\sqrt{x^2+y^2+z^2}) \end{bmatrix}$$

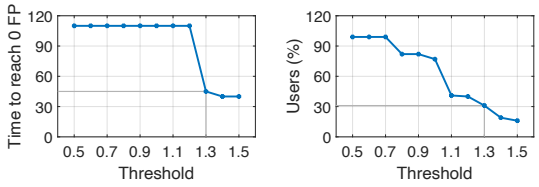
If we already have roll and pitch data from another sensor (e.g., IMU), the error function can become simpler. Instead of considering all three rotation angles, we can solve for yaw (γ). When the optimization is initiated with $\alpha_1 = \beta_1 = \gamma_1 = 0$, the first node sets the reference frame, and thus the orientations of all subsequent nodes are calculated relative to this frame.

3 Opportunistic Anchor Integration

Locate3D is fundamentally designed to function without the necessity for anchors. However, when anchors are available in the infrastructure, *Locate3D* can seamlessly integrate them to refine location and orientation results and transition from relative to global coordinates. This capability offers an edge over range-only systems, which strictly require at least three anchors for global coordinates, which is a challenging requirement in highly mobile environments. We categorize anchors into the following two types:

(a) Static anchors: They are conventional infrastructure anchors with known locations. These anchors have higher edge weightage, which increases their likelihood of being selected in MST over mobile volatile edges. Their impact on the accuracy is evaluated in a later section with varying anchor densities.

(b) Virtual anchors: As existing infrastructure cameras are available in many applications, we explore the potential to perform accurate localization in the vicinity and assign users within the camera's field-of-view as *virtual anchors*. A virtual anchor is a user with a high-confidence location obtained by an external modality, such as an infrastructure camera. It is important to note that while cameras can provide valuable localization information, they have limitations, such as reduced accuracy in low-light conditions.



(a) Time taken (in seconds) (b) Percent of users

Figure 8: (a) Time to reach '0 False positives'. (b) Percentage of users registered for varying trajectory matching thresholds.

User registration: Our registration technique uses correlation in user trajectories to incorporate human motion dynamics, such as varying walking speeds and stationary periods. The system benefits from *heading direction*, *cosine similarity of motion*, and speed analysis. We observe a scenario where some users are equipped with our node, while others are not, all within the camera's FoV. Our primary objective is to accurately register the system users in the camera's FoV while avoiding misclassifications. To address this, we adopt a stringent approach to trajectory similarity, aiming for zero false positives (FP). As depicted in Figure 8, keeping a higher threshold for registration effectively achieves zero FPs. By windowing 45 seconds of motion data, we register approximately 30% of visible users. It is important to note that while the availability of anchors can improve the accuracy, *Locate3D* does not rely on them for operation. We evaluate the system's performance under various anchor densities, to show its adaptability to different infrastructure setups.

4 Implementation

To this end, we implement a prototype of *Locate3D* and perform experiments in room-level, building-level, and city-level scenarios.

Node prototype: We developed prototype mobile nodes of *Locate3D*, as shown in Figure 11a to evaluate the performance. Each node consists of a Raspberry Pi 3 [63] for computation, an Intel Realsense T261 [36] to collect visual inertial odometry data. We use the NXP SR150 UWB evaluation boards [54] to collect the range, azimuth, and elevation AoA data. We collect the UWB measurements at a rate of 20Hz. All the nodes are time-synced using NTP.

Software: All the prototype nodes are part of a network and communicate with each other using a UDP socket. We

collect and store the UWB range, AoA, and VIO data from all nodes using Python scripts running on Raspberry Pi. We then process all measurements offline using Matlab where our algorithms are implemented.

5 Evaluation

We evaluate *Locate3D*'s performance in three key scenarios: room-scale, building-scale, and city-scale. In each scenario, we compare *Locate3D* with state-of-the-art baselines and benchmark localization accuracy, system latency, and the impact of various conditions such as line-of-sight (LOS) and non-line-of-sight (NLOS), stationary and moving nodes, lighting conditions, and varying numbers of infrastructure anchors. To assess *Locate3D*'s performance, we have collected over 4 hours of real-world data using 32 nodes and employed a motion capture system [3] as a ground truth setup to evaluate the centimeter-level localization accuracy and sub-degree-level orientation accuracy. We compare *Locate3D* with two distinct baseline systems: an infrastructure-free system, Capella [51], that uses UWB+VIO for peer-to-peer 3D localization, and an infrastructure-dependent system, ULoc [94], that uses UWB anchors and angular information for parallel node triangulation. Additionally, we simulate a large city-scale network by stitching real-world measurements to evaluate *Locate3D*'s scalability for networks with up to 100,000 nodes.

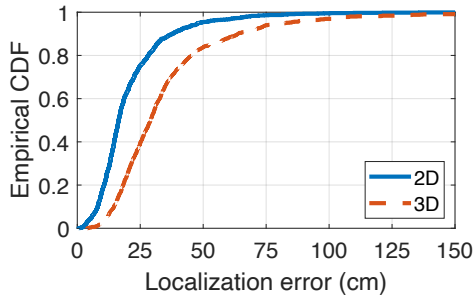


Figure 9: Room-scale evaluation: CDF of localization errors.

5.1 Localization Evaluation

5.1.1 Room-Scale Evaluation

We evaluate *Locate3D*'s performance in room-scale scenarios by conducting experiments in indoor rooms with dimensions of 6m $\times$ 8m and 4m $\times$ 20m, containing various furniture and equipment. We compare *Locate3D*'s performance with Cappella [51], a collaborative localization system that uses VIO for 3D tracking and inter-node UWB ranges to stitch individual trajectories using RBPF [80]. This evaluation involves 6 users walking naturally along random paths, emulating an AR/VR application, with additional users not participating in the system present to create a crowded and dynamic environment (Figure 11b). No anchors are used in these experiments as both *Locate3D* and Cappella are infrastructure-free. We use a Vicon motion capture system [83] as the ground truth, which provides sub-millimeter accuracy at a frame rate of 200Hz. The Vicon system consists of 12 Vantage V8 cameras mounted on the ceiling to track the ground truth locations and orientations of all nodes using active IR light and reflective markers attached to the nodes. We use 9 markers per node to ensure reliable tracking in all orientations. Figure 9 shows the CDF of the absolute localization error in 2D and 3D estimated by *Locate3D*, with median errors 18*cm* and 30*cm*, respectively, and 90th percentile errors of 38*cm* and 68*cm*.

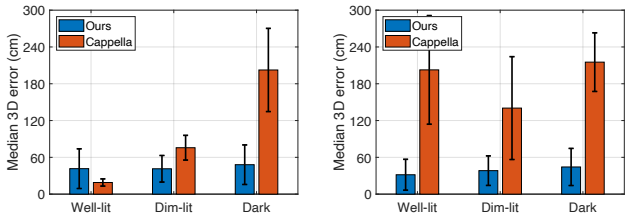


Figure 10: 3D Localization performance of *Locate3D* compared to baseline system - Cappella [51] - which uses visual odometry along with UWB. Results show performance in different lighting conditions for (a) Moving nodes and (b) Static nodes.

Effect of lighting: We evaluate *Locate3D*'s performance under varying lighting conditions to demonstrate its robustness to changes in illumination. As shown in Figure 10a, *Locate3D*'s performance remains consistent across different lighting conditions, while Cappella's performance degrades as indoor illumination decreases. The light intensity values for Well-lit, Dim-lit, and Dark conditions were 400, 150, 50 lux respectively which were controlled by analog light dimmers. This highlights *Locate3D*'s advantage of relying on RF-based peer-to-peer measurements, which are unaffected by visual conditions, in contrast to Cappella's reliance on VIO and visual features of the environment.

Effect of motion: Figure 10b shows that when nodes are static and the baseline system lacks a VIO trajectory tail, it fails to estimate 3D locations accurately. In contrast, *Locate3D* does not rely on odometry and can estimate the complete 6DOF location and orientation for all nodes, regardless of whether they are static or mobile.

5.1.2 Building-Scale Evaluation

We evaluate *Locate3D*'s performance in a multi-level building scenario using 32 nodes across multiple floors, as shown in Figure 11c, 11d. The experiments include a combination of LOS and NLOS conditions, with users walking in the building while holding nodes that capture images of calibrated AprilTags [87] placed on the walls for ground truth reference (Figure 11f). *Locate3D* achieves a median 3D localization error of 0.86 meters and an average orientation error of 4.5 degrees without using any infrastructure anchors. Figure 11e shows a snapshot of the estimated topology. To validate

(a) (b) (c) (d) (e) (f)

Figure 11: (a) Prototype built using Raspberry Pi, UWB sensor, Intel Realsense, IR markers for ground truth and a battery pack. (b) Room-scale evaluation. (c) Building-scale evaluation (d) 3D Lidar scan of the building for reference (not used in computation) (e) Snapshot of estimated locations and MST. (f) AprilTags [87] captured by nodes for ground truth (not used in computation).

the accuracy of the AprilTag-based ground truth system, we compared it against the Motion capture setup and found a 3D localization error of 0.11 meters. These results show *Locate3D*'s ability to provide accurate localization and orientation estimation in larger-scale, multi-level building environments without relying on infrastructure anchors. In section 5.2, we also do an ablation study to understand the contribution of each submodule to the system's latency and accuracy improvements.

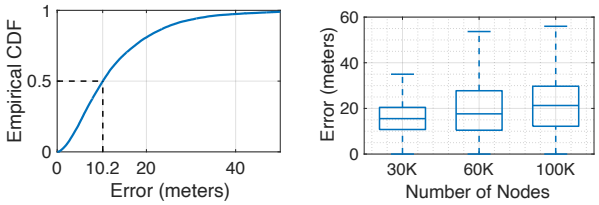


Figure 12: City-scale analysis: (a) CDF of localization errors of 30k nodes, 3800$m \times 3800m$ area, and 15 anchors. (b) Errors for 30k, 60k, and 100k nodes in 1500$m \times 1500m$ area and 1 anchor.

5.1.3 City-Scale Evaluation (Simulation)

We evaluate the performance of *Locate3D* in city-scale scenarios by simulating an environment with real-world collected data stitched in the simulation. The simulation incorporates real-world peer-to-peer measurements collected using 32 nodes in various environments, including indoor, outdoor, LOS, and NLOS scenarios. The dataset consists of 20,000 measurements covering ranges from 0.1 to 50 meters and AoA spanning the full 360 degrees from both static and mobile users. To emulate large-scale city networks with up to 100,000 nodes, we employ a data-driven approach by stitching together real-world measurements to preserve the noise properties. The simulation environment includes obstacles that mimic real-world urban settings, and the corresponding measurements are selected from the appropriate subset of the dataset (e.g., nodes 20m apart, 30° and 200° azimuth and elevation AoAs and NLOS edge) to maintain the integrity of the simulation. During the simulation process, we strictly adhere to the real-world measurements, ensuring that the nodes experience realistic noise and variations in range and AoA based on the specific distance and angle of the interaction. This enables us to evaluate *Locate3D*'s performance in large-scale networks while maintaining a strong connection to real-world conditions. Figure 12 shows the CDF of localization errors for 30k nodes using 15 anchors in a 3800$m \times 3800m$ area and the localization errors for 30k, 60k, and 100k nodes in a 1500$m \times 1500m$ area using 1 anchor (for global translation only). *Locate3D* achieves a median localization error of 12.09 meters for large-scale networks (100,000 nodes with 15 anchors) and 21 meters with 1 anchor.

We further evaluate *Locate3D* in a wide-area simulation of New York City, spanning approximately 22000$m \times 3200m$. We use 100k nodes with ground truth locations from the NYC transportation dataset [59], representing a Tuesday evening scenario. Figure 13 shows the results for configurations with 1 and 5 anchors. *Locate3D* achieved median localization errors of 82.31m and 34.19m for 1 and 5 anchors, respectively. The improved accuracy with more anchors demonstrates the system's ability to propagate location information effectively across wide-area environments.

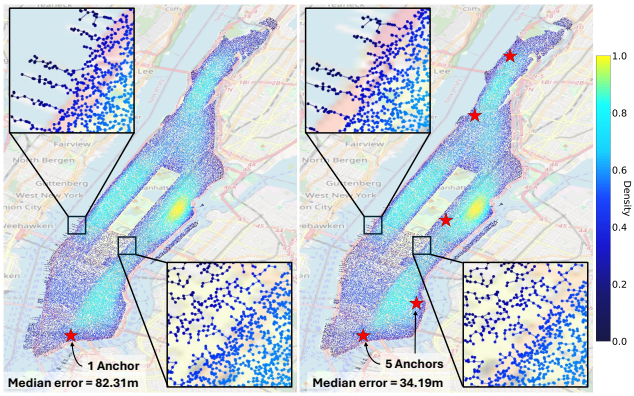
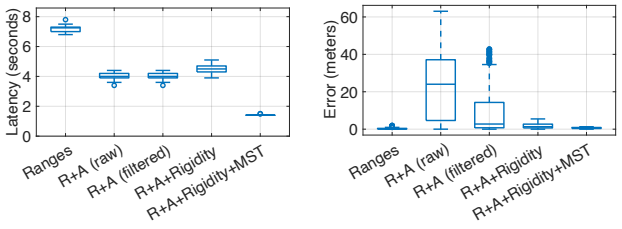


Figure 13: New York City wide-area simulation results: CDF of localization errors for 100,000 nodes using 1 (left figure) and 5 (right figure) anchors in a 22000$m \times 3200m$ area.

5.2 Latency Evaluation

To evaluate the effectiveness of each submodule in latency improvement, we conduct an experimental ablation study in the building-scale scenario. The results are shown in Figure 14. Starting with a baseline system that uses only range information, we observe that the introduction of

angle information (Range+Angle) significantly reduces the latency. However, the raw angle data leads to an increase in localization error due to the erroneous angles, as discussed in the section 2.2.2. To address this issue, we apply our filtering technique to fix erroneous angles which improves the accuracy while maintaining the latency reduction achieved by the introduction of angle information. The incorporation of rigidity constraints (R+A+Rigidity) further enhances the localization accuracy at a slight increase in latency compared to the filtered range and angle system. This is due to the addition of the critical edges that ensure a unique and unambiguous realization of the graph, as discussed in section 2.2.3. Finally, the optimal edge selection algorithm (R+A+MST) yields the most significant latency reduction while achieving localization accuracy comparable to the baseline system. This algorithm selects the optimal set of edges to minimize the overall latency while maintaining the graph's rigidity and uniqueness.



(a) (b)

Figure 14: Impact of submodules on (a) latency and (b) accuracy.

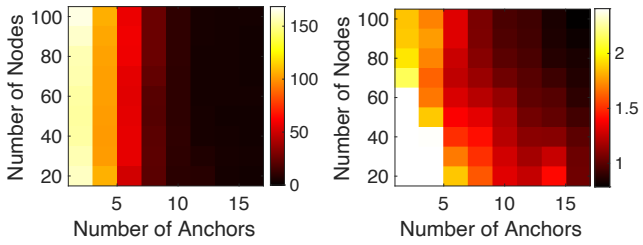
5.3 Micro-Benchmarks

5.3.1 Comparison with Infrastructure-Based Systems

To compare *Locate3D* with infrastructure-based systems, we evaluate the accuracy and scalability of *Locate3D* with a baseline infrastructure-based system ULoc [94], while varying the area, number of nodes, and number of anchors Our results show that *Locate3D* outperforms infrastructure-based systems in two key scenarios:

Sparse anchor deployments: *Locate3D*'s peer-to-peer network enables localization even when nodes are beyond the range of anchors, while ULoc fails in such cases. Figure 15 shows that both ULoc and *Locate3D* achieve meter-level errors when the number of anchors is high. However, as the number of anchors decreases, ULoc's average localization error increases due to nodes being out of range. Figure 16a demonstrates that with a fixed number of anchors and increasing area space, *Locate3D* can localize almost all nodes using its spanning tree, while the infrastructure-dependent technique loses connections to nodes making them unreachable.

NLOS edges: *Locate3D*'s edge selection algorithm prioritizes LOS measurements over NLOS, optimizing for overall noise reduction in the graph. In contrast, ULoc, being an infrastructure-based system, is constrained to using



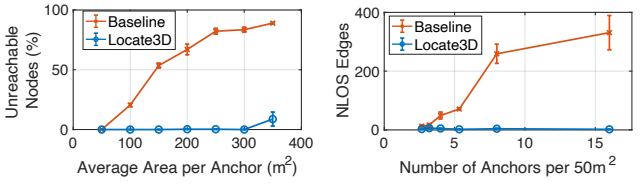
(a) Baseline (Infrastructure based) (b) *Locate3D*

Figure 15: Localization errors (in meters) for infrastructure baseline [94] and *Locate3D* for varying nodes and anchors.

anchor-to-node measurements regardless of whether they are LOS or NLOS. This limitation leads to the inclusion of noisy edges in the graph, affecting the overall localization performance. As shown in Figure 16b, *Locate3D* selectively includes LOS edges and minimizes the number of NLOS edges resulting in improved accuracy in realistic environments. By reducing the overall noise in the topology, *Locate3D* is able to provide more reliable and precise location estimates compared to ULoc, which is susceptible to errors introduced by NLOS measurements between nodes and anchors. This highlights the benefits of a peer-to-peer approach, which allows for greater flexibility in edge selection and noise reduction compared to infrastructure-based systems. However, infrastructure-based systems outperforms *Locate3D* in the following scenarios:

High anchor density and small-area experiments: As shown in Figure 15, infrastructure-based systems outperform in high anchor density cases where the error decreases due to close proximity with multiple anchors. Such small-scale scenarios with clear LOS and at least three anchors are where ULoc achieves better accuracy.

Locate3D's scope is to scale to a large number of nodes, while infrastructure-based systems are well-suited for small-scale deployments with sufficient anchor density.



(a) (b)

Figure 16: (a) Number of unreachable nodes and (b) Total number of NLOS measurements made with varying anchor area density.

5.3.2 LOS vs NLOS nodes

Figure 17(a) presents the CDF of the 2D and 3D localization errors in both line-of-sight and non-line-of-sight scenarios. In the NLOS scenario, there is a slight increase in localization error compared to the LOS scenario. This increase is primarily attributed to the higher noise levels in the received signals at the nodes due to obstructed signal paths and reflections. Specifically, in the 2D scenario, the median localization error

rises from 13*cm* in LOS to 28*cm* in NLOS. Similarly, in the 3D scenario, the median localization error increases from 31*cm* in LOS to 39*cm* in NLOS.

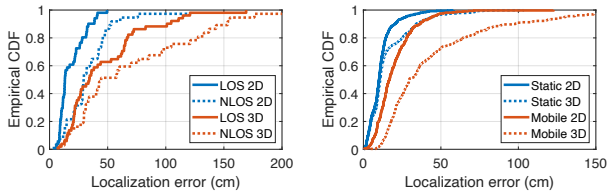
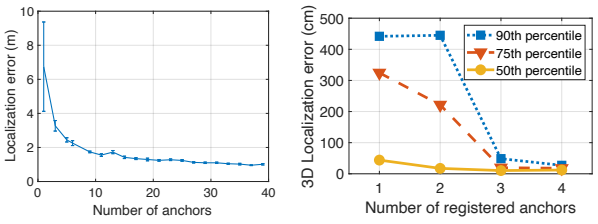


Figure 17: (a) LOS and NLOS localization errors. (b) Static vs Mobile nodes localization errors

5.3.3 Static vs Mobile nodes

To assess the impact of node mobility on the performance of *Locate3D*, we conduct experiments to compare static and mobile node scenarios. Figure 17(b) presents the CDF of 2D and 3D localization errors in both static and mobile node configurations. In the mobile scenario, the system experiences a slight degradation in performance due to varying environments and node vibrations. In the 2D scenario, the median localization error increases from a low of 6*cm* in the static setup to 13*cm* in the mobile scenario. Similarly, in the 3D scenario, the median localization error rises from 6.5*cm* in the static scenario to 26*cm* in the mobile scenario.



(a) (b)

Figure 18: (a) City-scale results for a 1000 node topology simulation in a $200 \times 200 \times 50$ meter 3D space for a varying number of static anchors. (b) Room-scale localization error for real-world 20-node experiments with varying numbers of anchors registered.

5.3.4 Impact of Anchor Density

We evaluate the impact of anchor density on localization performance in both large-scale simulations and real-world room-scale experiments. While *Locate3D* does not require anchors for operation, the presence of anchors can improve absolute localization accuracy in relative global coordinates. In a large-scale simulation with 1,000 nodes distributed within a 200m $\times$ 200m $\times$ 50m 3D space, *Locate3D* achieves a localization error below 2 meters with just 5 anchors (Figure 18a). In a real-world room-scale experiment, we evaluate the localization error while varying the number of registered anchors. *Locate3D* can peg the entire topology using only two anchors by leveraging angle information alongside location estimates. Figure 18b shows that the localization error decreases as more anchors are registered, with the main source of error being the rotation of the topology when only one anchor is present.

5.3.5 Ranging and AoA Performance

The localization errors inherently depend on the raging and AoA accuracies. We report the ranging and AoA estimation performance of the UWB sensor used in our implementation. While our system can be applied to any technology, here in the implementation we have used the UWB radio to get the distance and AoA error. So in this section, we report the accuracy of the ranging and angle estimation accuracy.
Figure 19 shows the ranging and AoA performance of our node for varying distances and incoming angles. We see that there are biases in certain angles as it is likely due to multipath. But still the 90^{th} percentile error remain within $\pm 5^{\circ}$.

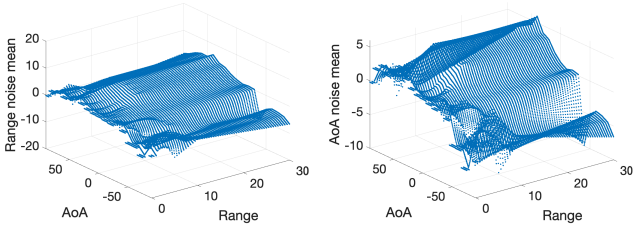


Figure 19: Ranging and AoA errors for various range-angles.

5.3.6 Orientation Estimation Accuracy

Figure 20(a) presents the CDF of the error in orientation estimation results. It is highly important for our system as these orientations play a critical role in transforming AoAs from the local frame of reference of the nodes to the global frame of reference, which is then used localization of nodes within a 3D space. Figure 20(a) shows that *Locate3D* consistently estimates accurate node's orientation with a median error of 4 degrees which is competitive with an inertial sensor.

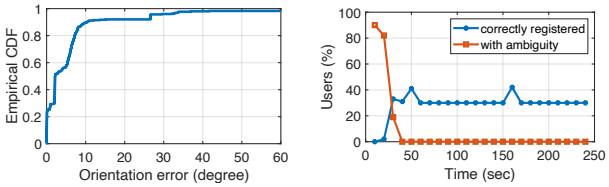


Figure 20: (a) CDF orientation errors. (b) Percent of users registered as Virtual anchors over time.

5.3.7 Virtual Anchor Registration

We evaluate the virtual anchor registration technique through a building-scale experiment with a network-connected camera. This setup included four node users and three non-participants walking within the camera's FoV, simulating typical crowd dynamics. Figure 20(b) shows the results that within the initial 40 seconds, the system registered 40% of the users as virtual anchors, while distinguishing between system users and non-participants.

6 Discussion

Scalability in real-time implementation: An extension of the proposed system with a semi-distributed approach can further improve its worst-case latency and robustness. In this case, the network is divided into sub-graphs managed by a local leader node. The leader nodes compute MSTs per sub-graphs in parallel and later aggregate to form the complete graph. This distributed architecture can offer better handling of high mobility scenarios subjected to future research.

Bootstrapping: *Locate3D*'s initial bootstrapping phase establishes the first set of measurements and constructs the initial graph without assumptions about node locations or network topology. Each node measures all neighboring nodes, building the graph incrementally based on responses. While this cold start phase naturally adds to the latency, it does not compromise localization accuracy. The duration of this phase depends on factors such as the number of nodes and application requirements. In subsequent iterations, *Locate3D* uses previous iteration information to efficiently select edges, reducing latency. While cold start is a common issue in state-of-the-art peer-to-peer localization systems, optimizing the bootstrapping process can be a direction for future work.

Applications: *Locate3D* caters to a wide range of applications requiring accurate and fast localization in infrastructure-free networks. Inspired by first responders scenarios, real-world experiments in multi-level buildings demonstrate the system's potential in emergency response scenarios. *Locate3D*'s ability to operate in GPS-denied environments and adapt to changes in network topology makes it suitable for disaster management and rescue operations. Its scalability, demonstrated through our city-scale simulations, shows its potential for smart city applications such as asset tracking, crowd monitoring, and urban planning. *Locate3D*'s framework can be easily adapted to incorporate application-specific parameters and constraints.

Advancements in angle field-of-view: Future sensor advancements, such as increased antenna counts, may enable 360° field-of-view coverage. In such cases, the spanning tree algorithm can be adapted to consider either range+angle edges or no edges, and account for varying noise distributions across different incoming angles. *Locate3D*'s framework is flexible to accommodate these advancements and generate optimal spanning trees without significant modifications.

Subgraphs without proximity for extended periods: Our critical edges technique addresses cases where subgraphs are out of vicinity for long durations. However, prolonged reliance can lead to drifting. Reinitializing the tree or cold-start can mitigate this issue. Presence of anchors in subgraphs can also provide reference points to align relative topology.

Other RF modalities: Our framework is adaptable to other modalities beyond UWB, including RF and acoustic sensors that can measure pairwise ranges and AoA [47, 48, 62, 88].

7 Related Work

Infrastructure-based systems: Motion tracking cameras (OptiTrack [1], Vicon [3]), passive markers (ARTags [41, 55], AprilTags [39, 87, 93]), and beacon-based solutions (UWB [57, 60, 75], Bluetooth [13, 43, 74], ultrasound [27, 43, 45]) are used for localization. These methods often combine ranging with odometry from IMU or VIO [16, 26, 49, 60, 69, 77, 85]. However, infrastructure-based systems are limited to prepared environments. *Locate3D* leverages peer-to-peer UWB ranges and angles, eliminating the need for infrastructure.

Infrastructure-free systems: Infrastructure-free localization determines relative positions between users without global references. This has been explored in sensor networks for stationary [56, 70] and mobile [18, 20, 53, 67] nodes, and in robotics using visual object detection [58, 84, 96] or odometry and distance measurements [6, 12, 31, 32, 46, 91, 92]. Recent research [51, 91, 92] employs visual-inertial features and UWB ranges for localization, but have limited scalability.

UWB localization: UWB-based systems have shown decimeter-level accuracy [25, 29, 30, 35, 40]. However, these systems often require a high number of infrastructure anchors, limiting their scalability. ULoc [94] uses AoA for triangulation but requires dense anchor deployments. XRLoc [5] employs a single large specialized anchor with multiple UWB nodes, focusing on small-scale, meter-level applications. SALMA [29] uses only one anchor but is limited to 2D and supports a small number of concurrent tags. In contrast, *Locate3D* is an infrastructure-free system that leverages both range and angle information for 6DoF localization. Unlike Cappella [51], which relies on VIO and UWB measurements, *Locate3D*'s performance is independent of lighting conditions. Moreover, *Locate3D* achieves localization without depending on odometry, preventing error accumulation.

8 Conclusion

This paper presents *Locate3D*, a peer-to-peer 3D localization and orientation estimation system for large networks. Our novel optimization and edge selection algorithm integrates both range and angle-of-arrival measurements for enhanced network topology estimation, demanding fewer edge measurements. As a result, *Locate3D* improves the latency by 75% without sacrificing accuracy, outperforming conventional range-only solutions.

Acknowledgement

This work was partially supported by NSF CAREER Award 2238433. We also thank the various companies that sponsor the iCoSMoS laboratory at UMD.

References

[1] Optitrack. https://optitrack.com/. Last accessed 12 March 2023.

[2] Technology and systems for drone shows. https://www.verge.aero/. Last accessed 21 September 2023.

[3] Vicon. https://www.vicon.com/. Last accessed 12 March 2023.

[4] Nima Alam and Andrew G Dempster. Cooperative positioning for vehicular networks: Facts and future. *IEEE transactions on intelligent transportation systems*, 14(4):1708–1717, 2013.

[5] Aditya Arun, Shunsuke Saruwatari, Sureel Shah, and Dinesh Bharadia. Xrloc: Accurate uwb localization to realize xr deployments. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, pages 459–473, 2023.

[6] Nan Bai, Yuan Tian, Ye Liu, Zhengxi Yuan, Zhuoling Xiao, and Jun Zhou. A high-precision and low-cost imu-based indoor pedestrian positioning technique. *IEEE Sensors Journal*, 20(12):6716–6726, 2020.

[7] Yang Bai, Nakul Garg, and Nirupam Roy. Spidr: Ultra-low-power acoustic spatial sensing for micro-robot navigation. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 99–113, 2022.

[8] Yang Bai, Nakul Garg, and Nirupam Roy. Microstructure-assisted vision: Adding new senses to low-power devices. *GetMobile: Mobile Computing and Communications*, 27(4):15–20, 2024.

[9] Brian Beck, Robert Baxley, and Joseph Kim. Real-time, anchor-free node tracking using ultrawideband range and odometry data. In *2014 IEEE international conference on ultra-wideband (ICUWB)*, pages 286–291. IEEE, 2014.

[10] Neha Bhadwal, Vishu Madaan, Prateek Agrawal, Awadesh Shukla, and Anuj Kakran. Smart border surveillance system using wireless sensor network and computer vision. In *2019 international conference on Automation, Computational and Technology Management (ICACTM)*, pages 183–190. IEEE, 2019.

[11] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3-4):561–575, 2003.

[12] Zhiqiang Cao, Ran Liu, Chau Yuen, Achala Athukorala, Benny Kai Kiat Ng, Muraleetharan Mathanraj, and U-Xuan Tan. Relative localization of mobile robots with multiple ultra-wideband ranging measurements. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5857–5863. IEEE, 2021.

[13] Srdjan Čapkun, Maher Hamdi, and Jean-Pierre Hubaux. Gps-free positioning in mobile ad hoc networks. *Cluster Computing*, 5:157–167, 2002.

[14] Hongyang Chen, Gang Wang, Zizhuo Wang, Hing-Cheung So, and H Vincent Poor. Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 11(1):108–116, 2011.

[15] Jan De Leeuw and Patrick Mair. Multidimensional scaling using majorization: Smacof in r. *Journal of statistical software*, 31:1–30, 2009.

[16] Ashutosh Dhekne, Ayon Chakraborty, Karthikeyan Sundaresan, and Sampath Rangarajan. Trackio: Tracking first responders inside-out. In *NSDI*, volume 19, pages 751–764, 2019.

[17] Carmelo Di Franco, Amanda Prorok, Nikolay Atanasov, Benjamin Kempke, Prabal Dutta, Vijay Kumar, and George J Pappas. Calibration-free network localization using non-line-of-sight ultra-wideband measurements. In *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 235–246, 2017.

[18] Tolga Eren, OK Goldenberg, Walter Whiteley, Yang Richard Yang, A Stephen Morse, Brian DO Anderson, and Peter N Belhumeur. Rigidity, computation, and randomization in network localization. In *IEEE INFOCOM 2004*, volume 4, pages 2673–2684. IEEE, 2004.

[19] Fira. Unleashing the potential of uwb: Regulatory considerations. *FiRa Consortium*, August 2022.

[20] Nakul Garg, Yang Bai, and Nirupam Roy. Owlet: Enabling spatial information in ubiquitous acoustic devices. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21), June 24–July 2, 2021, Virtual, WI, USA*. ACM.

[21] Nakul Garg, Aritrik Ghosh, and Nirupam Roy. Litefoot: Ultra-low-power localization using ambient cellular signals. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2024)*, New York, NY, USA, 2024. ACM. University of Maryland, College Park.

[22] Nakul Garg and Nirupam Roy. Enabling self-defense in small drones. In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*, pages 15–20, 2020.

[23] Nakul Garg and Nirupam Roy. poster: Ultra-low-power angle-of-arrival estimation using a single antenna. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 569–570, 2023.

[24] Nakul Garg and Nirupam Roy. Sirius: A self-localization system for resource-constrained iot sensors. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 289–302, 2023.

[25] Nakul Garg, Irtaza Shahid, Karthik Sankar, Mallesham Dasari, Ramanujan K Sheshadri, Karthikeyan Sundaresan, and Nirupam Roy. Bringing ar/vr to everyday life-a wireless localization perspective. In *Proceedings of the 24th International Workshop on Mobile Computing Systems and Applications*, pages 142–142, 2023.

[26] Christian Gentner and Markus Ulmschneider. Simultaneous localization and mapping for pedestrians using low-cost ultra-wideband system and gyroscope. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2017.

[27] David Gómez, Paula Tarrío, Juan Li, Ana M Bernardos, and José R Casar. Indoor augmented reality based on ultrasound localization systems. In *Highlights on Practical Applications of Agents and Multi-Agent Systems: International Workshops of PAAMS 2013, Salamanca, Spain, May 22-24, 2013. Proceedings 11*, pages 202–212. Springer, 2013.

[28] Ronald L Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57, 1985.

[29] Bernhard Großwindhager, Michael Rath, Josef Kulmer, Mustafa S Bakr, Carlo Alberto Boano, Klaus Witrisal, and Kay Römer. Salma: Uwb-based single-anchor localization system using multipath assistance. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 132–144, 2018.

[30] Bernhard Großwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer. Snaploc: An ultra-fast uwb-based indoor localization system for an unlimited number of tags. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 61–72, 2019.

[31] Kexin Guo, Xiuxian Li, and Lihua Xie. Ultra-wideband and odometry-based cooperative relative localization with application to multi-uav formation control. *IEEE transactions on cybernetics*, 50(6):2590–2603, 2019.

[32] Kexin Guo, Zhirong Qiu, Wei Meng, Lihua Xie, and Rodney Teo. Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments. *International Journal of Micro Air Vehicles*, 9(3):169–186, 2017.

[33] Moshe Hamaoui. Non-iterative mds method for collaborative network localization with sparse range and pointing measurements. *IEEE Transactions on Signal Processing*, 67(3):568–578, 2018.

[34] Yong-Gu Han, Youngho Cho, Ohseok Kwon, et al. The use of conservation drones in ecology and wildlife research. *Journal of Ecology and Environment*, 38(1):113–118, 2015.

[35] Frank Hartmann, Felix Pistorius, Andreas Lauber, Kai Hildenbrand, Jürgen Becker, and Wilhelm Stork. Design of an embedded uwb hardware platform for navigation in gps denied environments. In *2015 IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, pages 1–6. IEEE, 2015.

[36] Intel. https://dev.intelrealsense.com/docs/tracking-camera-t265-datasheet, 2023.

[37] Vikram Iyer, Hans Gaensbauer, Thomas L Daniel, and Shyamnath Gollakota. Wind dispersal of battery-free wireless devices. *Nature*, 603(7901):427–433, 2022.

[38] Vikram Iyer, Maruchi Kim, Shirley Xue, Anran Wang, and Shyamnath Gollakota. Airdropping sensor networks from drones and insects. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[39] Jan Kallwies, Bianca Forkel, and Hans-Joachim Wuensche. Determining and improving the localization accuracy of apriltag detection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8288–8294. IEEE, 2020.

[40] Benjamin Kempke, Pat Pannuto, Bradford Campbell, and Prabal Dutta. Surepoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 137–149, 2016.

[41] Manfred Klopschitz and Dieter Schmalstieg. Automatic reconstruction of wide-area fiducial marker models. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 71–74. IEEE, 2007.

[42] Gerard Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering mathematics*, 4(4):331–340, 1970.

[43] Patrick Lazik, Niranjini Rajagopal, Oliver Shih, Bruno Sinopoli, and Anthony Rowe. Alps: A bluetooth and ultrasound platform for mapping and localization. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pages 73–84, 2015.

[44] Inhee Lee, Roger Hsiao, Gordy Carichner, Chin-Wei Hsu, Mingyu Yang, Sara Shoouri, Katherine Ernst, Tess Carichner, Yuyang Li, Jaechan Lim, et al. msail: milligram-scale multi-modal sensor platform for monarch butterfly migration tracking. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 517–530, 2021.

[45] Jinyang Li, Zhiheng Xie, Xiaoshan Sun, Jian Tang, Hengchang Liu, and John A Stankovic. An automatic and accurate localization system for firefighters. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 13–24. IEEE, 2018.

[46] Ming Li, Zhuang Chang, Zhen Zhong, and Yan Gao. Relative localization in multi-robot systems based on dead reckoning and uwb ranging. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–7. IEEE, 2020.

[47] Fen Liu, Jing Liu, Yuqing Yin, Wenhan Wang, Donghai Hu, Pengpeng Chen, and Qiang Niu. Survey on wifi-based indoor positioning techniques. *IET communications*, 14(9):1372–1383, 2020.

[48] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. Push the limit of wifi based localization for smartphones. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 305–316, 2012.

[49] Ran Liu, Chau Yuen, Tri-Nhut Do, Dewei Jiao, Xiang Liu, and U-Xuan Tan. Cooperative relative positioning of mobile users by fusing imu inertial and uwb ranging information. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5623–5629. IEEE, 2017.

[50] Volodymyr Melnykov and Igor Melnykov. Initializing the em algorithm in gaussian mixture models with an unknown number of components. *Computational Statistics & Data Analysis*, 56(6):1381–1395, 2012.

[51] John Miller, Elahe Soltanaghai, Raewyn Duvall, Jeff Chen, Vikram Bhat, Nuno Pereira, and Anthony Rowe. Cappella: Establishing multi-user augmented reality sessions using inertial estimates and peer-to-peer ranging. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 428–440. IEEE, 2022.

[52] Daniel Minoli and Benedict Occhiogrosso. Ultrawideband (uwb) technology for smart cities iot applications. In *2018 IEEE international smart cities conference (ISC2)*, pages 1–8. IEEE, 2018.

[53] David Moore, John Leonard, Daniela Rus, and Seth Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, 2004.

[54] Mouser. https://www.mouser.com/new/murata/murata-type2bp-uwb-module-eval-kit/, 2024.

[55] Alessandro Mulloni, Hartmut Seichter, and Dieter Schmalstieg. Handheld augmented reality indoor navigation with activity-based instructions. In *Proceedings of the 13th international conference on human computer interaction with mobile devices and services*, pages 211–220, 2011.

[56] Radhika Nagpal, Howard Shrobe, and Jonathan Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Information Processing in Sensor Networks: Second International Workshop, IPSN 2003, Palo Alto, CA, USA, April 22–23, 2003 Proceedings*, pages 333–348. Springer, 2003.

[57] Thien Hoang Nguyen, Thien-Minh Nguyen, and Lihua Xie. Range-focused fusion of camera-imu-uwb for accurate and drift-reduced localization. *IEEE Robotics and Automation Letters*, 6(2):1678–1685, 2021.

[58] Ty Nguyen, Kartik Mohta, Camillo J Taylor, and Vijay Kumar. Vision-based multi-mav localization with anonymous relative measurements using coupled probabilistic data association filter. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3349–3355. IEEE, 2020.

[59] NYC Metropolitan Transportation Authority. Nyc mta turnstile data, 2024. [Online]. Available: https://data.ny.gov/browse/select_dataset?category=Transportation.

[60] Fredrik Olsson, Jouni Rantakokko, and Jonas Nygårds. Cooperative localization using a foot-mounted inertial navigation system and ultrawideband ranging. In *2014 International conference on indoor positioning and indoor navigation (IPIN)*, pages 122–131. IEEE, 2014.

[61] Zhongmin Pei, Zhidong Deng, Shuo Xu, and Xiao Xu. Anchor-free localization method for mobile targets in coal mine wireless sensor networks. *Sensors*, 9(4):2836–2850, 2009.

[62] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14, 2007.

[63] Raspberry Pi. https://www.raspberrypi.com/products/raspberry-pi-3-model-b/, 2023.

[64] Yihong Qi. *Wireless geolocation in a non-line-of-sight environment*. Princeton University, 2003.

[65] Qorvo. https://www.mouser.com/datasheet/2/412/DWM1000_Data_Sheet-1950396.pdf, 2016.

[66] Qorvo. https://www.qorvo.com/products/d/da008334, 2021.

[67] Hadi Jamali Rad, Alon Amar, and Geert Leus. Cooperative mobile network localization via subspace tracking. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2612–2615. IEEE, 2011.

[68] Fatma Raissi, Sami Yangui, and Frederic Camps. Autonomous cars, 5g mobile networks and smart cities: Beyond the hype. In *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 180–185. IEEE, 2019.

[69] Niranjini Rajagopal, John Miller, Krishna Kumar Reghu Kumar, Anh Luong, and Anthony Rowe. Improving augmented reality relocalization using beacons and magnetic field maps. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2019.

[70] Andreas Savvides, Chih-Chieh Han, and Mani B Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, 2001.

[71] Judy Scott and Carlton Scott. Drone delivery models for healthcare. 2017.

[72] Yi Shang, Wheeler Rumi, Ying Zhang, and Markus Fromherz. Localization from connectivity in sensor networks. *IEEE Transactions on parallel and distributed systems*, 15(11):961–974, 2004.

[73] Yi Shang and Wheeler Ruml. Improved mds-based localization. In *IEEE INFOCOM 2004*, volume 4, pages 2640–2651. IEEE, 2004.

[74] Chong Shao, Bashima Islam, and Shahriar Nirjon. Marble: Mobile augmented reality using a distributed ble beacon infrastructure. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 60–71. IEEE, 2018.

[75] Wang Shule, Carmen Martínez Almansa, Jorge Peña Queralta, Zhuo Zou, and Tomi Westerlund. Uwb-based localization for multi-uav systems and collaborative heterogeneous multi-robot systems. *Procedia Computer Science*, 175:357–364, 2020.

[76] Gregory G Slabaugh. Computing euler angles from a rotation matrix. *Retrieved on August*, 6(2000):39–63, 1999.

[77] Yang Song, Mingyang Guan, Wee Peng Tay, Choi Look Law, and Changyun Wen. Uwb/lidar fusion for cooperative range-only slam. In *2019 international conference on robotics and automation (ICRA)*, pages 6568–6574. IEEE, 2019.

[78] Francesco Sottile, Maurizio A Spirito, Mauricio A Caceres, and Jaron Samson. Distributed-weighted multidimensional scaling for hybrid peer-to-peer localization. In *2010 Ubiquitous Positioning Indoor Navigation and Location Based Service*, pages 1–6. IEEE, 2010.

[79] Benjamin Thompson and Noah Baker. The floating sensors inspired by seeds. *Nature*, 2021.

[80] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[81] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. {FarmBeats}: an {IoT} platform for {Data-Driven} agriculture. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 515–529, 2017.

[82] Saipradeep Venkatraman, James Caffery, and Heung-Ryeol You. A novel toa location algorithm using los range estimation for nlos environments. *IEEE Transactions on Vehicular Technology*, 53(5):1515–1524, 2004.

[83] Vicon. https://docs.vicon.com/display/Vantage/V8+camera+specifications, 2023.

[84] Viktor Walter, Nicolas Staub, Antonio Franchi, and Martin Saska. Uvdar system for visual relative localization with application to leader–follower formations of multirotor uavs. *IEEE Robotics and Automation Letters*, 4(3):2637–2644, 2019.

[85] Chen Wang, Handuo Zhang, Thien-Minh Nguyen, and Lihua Xie. Ultra-wideband aided fast localization and mapping system. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1602–1609. IEEE, 2017.

[86] Jiadai Wang, Jiajia Liu, and Nei Kato. Networking and communications in autonomous driving: A survey. *IEEE Communications Surveys & Tutorials*, 21(2):1243–1274, 2018.

[87] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE, 2016.

[88] Pengfei Wang and Yufeng Luo. Research on wifi indoor location algorithm based on rssi ranging. In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pages 1694–1698. IEEE, 2017.

[89] Eric W Weisstein. Euler angles. https://mathworld.wolfram.com/, 2009.

[90] Henk Wymeersch, Gabriel Rodrigues de Campos, Paolo Falcone, Lennart Svensson, and Erik G Ström. Challenges for cooperative its: Improving road safety through the integration of wireless communications, control, and positioning. In *2015 international conference on computing, networking and communications (ICNC)*, pages 573–578. IEEE, 2015.

[91] Hao Xu, Luqi Wang, Yichen Zhang, Kejie Qiu, and Shaojie Shen. Decentralized visual-inertial-uwb fusion for relative state estimation of aerial swarm. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 8776–8782. IEEE, 2020.

[92] Hao Xu, Yichen Zhang, Boyu Zhou, Luqi Wang, Xinjie Yao, Guotao Meng, and Shaojie Shen. Omni-swarm: A decentralized omnidirectional visual–inertial–uwb state estimation system for aerial swarms. *IEEE Transactions on Robotics*, 38(6):3374–3394, 2022.

[93] Boxin Zhao, Zongzhe Li, Jun Jiang, and Xiaolin Zhao. Relative localization for uavs based on april-tags. In *2020 Chinese Control And Decision Conference (CCDC)*, pages 444–449. IEEE, 2020.

[94] Minghui Zhao, Tyler Chang, Aditya Arun, Roshan Ayyalasomayajula, Chi Zhang, and Dinesh Bharadia. Uloc: Low-power, scalable and cm-accurate uwb-tag localization and tracking for indoor applications. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–31, 2021.

[95] Kan Zheng, Qiang Zheng, Haojun Yang, Long Zhao, Lu Hou, and Periklis Chatzimisios. Reliable and efficient autonomous driving: the need for heterogeneous vehicular networks. *IEEE Communications Magazine*, 53(12):72–79, 2015.

[96] Thomas Ziegler, Marco Karrer, Patrik Schmuck, and Margarita Chli. Distributed formation estimation via pairwise distance measurements. *IEEE Robotics and Automation Letters*, 6(2):3017–3024, 2021.