Communication-Driven Localization and Mapping for Millimeter Wave Networks

Joan Palacios[♯‡], Guillermo Bielsa[♯‡], Paolo Casari[♯], Joerg Widmer[♯]
[♯]IMDEA Networks Institute, Madrid, Spain [‡]Universidad Carlos III, Madrid, Spain

***Abstract*—Millimeter wave (mmWave) communications are an essential component of 5G-and-beyond ultra-dense Gbit/s wireless networks, but also pose significant challenges related to the communication environment. Especially beam-training and tracking, device association, and fast handovers for highly directional mmWave links may potentially incur a high overhead. At the same time, such mechanisms would benefit greatly from accurate knowledge about the environment and device locations that can be provided through simultaneous localization and mapping (SLAM) algorithms.**

In this paper we tackle the above issues by proposing CLAM, a distributed mmWave SLAM algorithm that works with no initial information about the network deployment or the environment, and achieves low computational complexity thanks to a fundamental reformulation of the angle-differences-of-arrival mmWave anchor location estimation problem. All information required by CLAM is collected by a mmWave device thanks to beam training and tracking mechanisms inherent to mmWave networks, at no additional overhead. Our results show that CLAM achieves submeter accuracy in the great majority of cases. These results are validated via an extensive experimental measurement campaign carried out with 60-GHz mmWave hardware.

I. Introduction

Millimeter wave (mmWave) communications in the 30–300 GHz band are considered key ingredients to achieve multiple Gbit/s link rates in 5G-and-beyond networks [1] as well as WLANs [2]. First mmWave devices following the IEEE 802.11ad standard are commercially available [3], [4].

mmWave signals follow quasi-optical propagation patterns, with clear reflections off boundary surfaces, and little scattering [5]. However, the high frequency of mmWave transmissions implies short coverage ranges. Moreover, mmWaves are blocked by a number of materials, including the human body [6]. Several materials reflect mmWave signals, providing alternative paths in case no line-of-sight (LoS) path is available. To achieve viable link ranges, mmWave devices employ directional, electronically steerable antenna arrays. The identification of the best steering direction for the antenna's main lobe is called beam training [7], [8]. For relatively static scenarios, brute force or hierarchical training as used, e.g., in the 802.11ad standard [9] work reasonably well. More refined methods exist that reduce beam training times, can track multiple paths at once, and work better in dynamic scenarios [10]–[12].

The quasi-optical propagation of mmWave signals, the sparse angle of arrival (AoA) spectrum that results, and the capability to track multiple components of the AoA spectrum, imply a very desirable consequence: that mmWave technology seamlessly allows mobile network devices to achieve simultaneous localization and mapping (SLAM) through appropriately designed AoA-based methods. Such information is instrumental to mmWave networks, as it enables context awareness, and can facilitate beam training or handover operations, and can be used for location-based services and applications [13]. For example, gathering enough AoA information for device localization allows to do immediate handovers without the need for beam training to *any* access point (AP) for which only the position is known. To be viable for practical mmWave systems, a SLAM algorithm:

1) should ideally be run locally by each client, without any additional special-purpose messaging (which allows it to run on any 5G or 802.11ad compliant devices);
2) should work with commercial off-the-shelf (COTS) devices, which are often low-cost, non-calibrated, and subject to computational constraints;
3) should not require to engineer or manually configure the network deployment (especially the AP locations).

The above constraints have profound implications on the design of a SLAM algorithm for mmWave networks. Specifically, to achieve requirement 1) no additional data should be exchanged among the devices, or between the device and the APs, requiring each device to achieve SLAM independently. Only existing information, i.e., AoAs extracted from beam training, can be leveraged for localization. Range-based mechanisms such as [14] are not viable on unmodified COTS hardware, and are thus incompatible with requirement 2). Moreover, the designed algorithm should not be computationally complex, so that it can run in real-time on low-end devices. Finally, due to 3) the AP locations are not known in advance and cannot be distributed to the network devices (which would also require special-purpose messaging). Requirement 3) also prevents fingerprinting-based algorithms, which would incur a very high network setup cost.

Our goal is to overcome the constraints and issues above, by designing a zero-initial information, zero-overhead, low-complexity SLAM algorithm. We approach the localization part of SLAM through angle difference of arrival (ADoA) information. This entails two sub-steps: estimating the locations of the surrounding APs, and deriving the position of the client based on AP locations. The former is achieved through a novel approach that finds and solves a minimal number of relationships among ADoA measurements obtained by a client as it moves throughout an indoor environment. These relationships are found offline through automatic expression manipulation techniques. After a sufficient number of APs

have been located, an error-resilient version of the ADoA localization algorithm is employed to estimate the client's position. All required ADoA information can be derived from the output of the beam training as carried out according to the standard. We remark that several approaches exist to track the LoS and non-line-of-sight (NLoS) components of a sparse mmWave AoA spectrum. Having NLoS information available makes it possible for our algorithm to work in the presence of realistic propagation issues, including blockage originating from the environment or from other mobile users.

In light of our zero-initial information constraint, our algorithm estimates all data needed for SLAM, including the location of physical APs (which are the actual sources of LoS arrivals) and of virtual APs (which are the virtual sources of NLoS arrivals, and can be modeled by mirroring the location of a physical AP through a surface that reflects the mmWave signal). This makes the problem much harder, but we show that it can still be tackled through ADoA information. Since typically LoS and NLoS arrivals from the same physical AP are available (although not necessarily at the same point in time), the coupling of virtual APs to their originating physical APs allows to estimate the location and shape of the boundaries of the environment and of the obstacles therein.

Finally, we need to limit the complexity of the location estimation algorithm. We achieve this via a fundamental reformulation of the ADoA joint AP and user localization algorithm, that is adapted to the challenging case of zero-initial information. Our formulation is amenable to fast initialization procedures to find initial estimates of the AP and device locations, as well as to low-complexity updating mechanisms for successive refinements of such estimates as the device moves. To the best of our knowledge, this formulation has never been used in the literature related to AoA-based localization approaches. We note that our method is very different from traditional SLAM approaches used, for example, in the field of robotics. There, SLAM is typically a *dedicated* mechanism whose primary objective is to make vehicles or robots aware of the environment; it is often achieved through radars, laser or cameras, by leveraging movement direction information supplied by the robot's sensors, and in the presence of more landmarks than anchors available in a typical 60 GHz WLAN. In contrast, our SLAM algorithm is *embedded* in the network, as it relies only on operations that are carried out for standard communications. The information it generates is instrumental to optimize the network behavior in an anticipatory fashion. Additionally, it complies with COTS hardware constraints, as it does not rely on radar-like approaches or on special-purpose equipment. Our specific contributions are:

- a new approach that tackles ADoA localization from a fundamental standpoint to design a distributed, zero-initial information, zero-overhead, low-complexity algorithm (Sections II and III);
- a thorough evaluation of our algorithm through experiments with 60-GHz mmWave hardware, backed by simulations for a more comprehensive investigation (Sections IV and V).

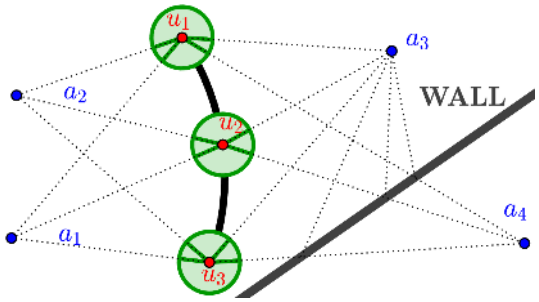


Fig. 1. Scenario with a single mobile user with four anchor nodes (blue, a_1 to a_4). Red dots at u_1 to u_3 are ADoA measurement locations. Circles represent both AoAs (radii) and ADoAs (spanned by arcs between radii pairs).

II. Anchor Position Estimation

Our algorithm for communication-driven localization and mapping in mmWave networks (CLAM) is composed of three parts: anchor location estimation, device localization, and environment mapping. Consider the scenario in Fig. 1, where a single mobile device receives signals from four anchor nodes, of which a_1, a_2 and a_3 are physical APs, whereas a_4 is the virtual AP that models the source of NLoS signals from a_3. We recall that a_4 is obtained by mirroring the location of a_3 through the reflecting wall. In subsequent locations u_1, u_2, and u_3, the user leverages beam training information to compute the ADoA between the signals from every anchor pair. The arcs in the circles centered on u_1, u_2, and u_3 in Fig. 1 visualize ADoAs, and circle radii convey the corresponding AoAs. We remark that CLAM works with physical and virtual anchors alike. As a result, multi-path mmWave propagation helps increase the localization accuracy.

CLAM estimates the locations of the anchor nodes starting from zero initial information. It does so in such a way that the anchor locations are compatible with ADoA measurements taken at different positions, corresponding to angle measurements extracted from the standard beam training mechanism. ADoA anchor localization is invariant to rotation, translation and scaling. We therefore consider the relative localization of the user and APs, and define a solution to the anchor localization process as the *equivalence class* of the valid anchor locations under the three above transformations.[1] We call this class an *anchor shape*, or simply a *shape*, and denote it with the symbol $\mathcal{S}$. For example, the anchor shape of anchors a_1 to a_4 in Fig. 1 is their set of coordinates, along with any rotation, translation and scaling of these coordinates. Note that knowing the shape is equivalent to knowing any of its elements, and all angles between them. In the following, we will denote a set of anchors as $\mathcal{A}$, and the shape that refers to this set of anchors as $\mathcal{S}_{\mathcal{A}}$.

The anchor shape can be estimated by observing different ADoAs from the same anchor set across different measurements taken at different user locations. The intuition is that a

[1]We remark that all ambiguities can be resolved by knowing the coordinates of any two points in the localization area, e.g., the coordinates of two APs. However, using the location information for network optimization such as beam tracking or fast AP handovers does not require such disambiguation. A device can base these decisions on its own local coordinate system.

given anchor shape $\mathcal{S}$ generates a set of possible ADoAs $\Gamma_{\mathcal{S}}$. This set univocally determines a shape. Using multiple ADoA sets, it is possible to detect a compatible shape by fusing the different measurements.

Note that the anchor position estimation problem is underdetermined if the user sees just three anchors. Formally, for a given anchor set $\mathcal{A}$, $\Gamma_{\mathcal{S}_\mathcal{A}}$ can be computed by mapping every point of the 2D space into the ADoAs Θ that a user would observe at that point. This defines a manifold of dimension 2 inside a space of dimension $|\mathcal{A}|-1$. Therefore, when $|\mathcal{A}| \geq 3$, the set of possible ADoAs $\Gamma_{\mathcal{S}_\mathcal{A}}$ is contained in, but does not span, the full space of angle combinations, and can thus be used to infer the shape $\mathcal{S}_\mathcal{A}$. Therefore, we develop the model by assuming that the user measures ADoAs from exactly four anchors, and then extend it to any number of anchors.

A. Anchor shape estimation

Our novel formulation of the ADoA anchor localization problem is based on determining a non-trivial implicit expression $\mathcal{R}(\mathcal{S},\Theta)$ that ties an anchor shape $\mathcal{S}$ to a set of measured ADoAs Θ, such that $\mathcal{R}(\mathcal{S},\Theta)=0$ if $\mathcal{S}$ is an anchor shape compatible with Θ. In contrast with previous work such as [15], [16], we do not directly exploit the geometry of the ADoA localization problem, but rather we inspect the relationship among the anchors involved, in order to find an underlying structure that can reveal the expression $\mathcal{R}(\mathcal{S},\Theta)$.

Call *anchor angles* the angles formed among any three anchors (e.g., $\widehat{a_1a_2a_3}$ and $\widehat{a_2a_3a_4}$ in Fig. 1), and let $\xi_{ijk} = |\widehat{a_ia_ja_k}|$ be the amplitude of the corresponding angle. Note that the anchor angles fully determine the anchor shape. The intuition behind the following development is that the relation $\mathcal{R}$ is a trigonometric polynomial of a certain order, whose terms are powers of trigonometric functions of the anchor angles and ADoAs. This relationship can be expressed as

$$\mathcal{R}(\mathcal{S},\Theta)=\sum_i\sum_j p_i m_{i,j} s_j = \mathbf{p}^T\mathbf{M}\mathbf{s} = \mathrm{vec}(\mathbf{M})^T(\mathbf{p}\otimes\mathbf{s}), \quad (1)$$

where each p_i and each s_j are trigonometric polynomial functions of the ADoAs and of the anchor angles, respectively; $\mathbf{p},\mathbf{s}$ denote their vector representation (we single out a linearly independent set of terms), $\mathbf{M}$ is the matrix of polynomial coefficients, $\mathrm{vec}(\,\cdot\,)$ denotes the column-wise vectorization of the argument, ^{T} denotes transposition and $\otimes$ is the Kronecker product. Call $\zeta_{ik}(u_j) = |\widehat{a_iu_ja_k}|$. An order-3 term would be for example $\sin\zeta_{13}(u_1)\cos\zeta_{24}(u_1)\sin\zeta_{24}(u_1)$ for ADoAs, or $\sin\xi_{123}\cos^2\xi_{234}$ for anchor angles.

If $\mathcal{R}(\Theta,\mathcal{S})$ exists and is non-trivial, we can generate a large number K of random shapes and locations and assemble the products $(\mathbf{p}\otimes\mathbf{s})_i$, $i=1,\dots,K$, into a matrix

$$\mathbf{X} = [(\mathbf{p}\otimes\mathbf{s})_1, (\mathbf{p}\otimes\mathbf{s})_2, \cdots (\mathbf{p}\otimes\mathbf{s})_K]\,. \quad (2)$$

This matrix satisfies $\mathrm{vec}(\mathbf{M})^T\mathbf{X} = \mathbf{0}_{1\times K}$. Therefore,

$$\mathrm{vec}(\mathbf{M})^T\mathbf{X}\mathbf{X}^T\mathrm{vec}(\mathbf{M})=0, \text{ or } \mathrm{vec}(\mathbf{M})\in\ker(\mathbf{X}\mathbf{X}^T)\backslash\{\mathbf{0}\} \quad (3)$$

where $\ker(\cdot)$ denotes the kernel of the argument.

We compute matrix $\mathbf{X}$ for different orders of the trigonometric polynomial functions of the ADoAs and anchor angles. Through automatic expression manipulation, we then prove that the minimum order that leads to a non-trivial kernel is 2 for the ADoA terms, and 3 for the anchors angle terms. This makes it possible to simplify the formulation by reducing the dimension of both $\mathbf{p}$ and $\mathbf{s}$ to 5, such that we finally have that $\mathcal{R}(\mathcal{S},\Theta) = \mathbf{p}^T\mathbf{s} = 0$ for every anchor shape $\mathcal{S}$ compatible with the set of measured ADoAs Θ. We remark that the generation of matrix $\mathbf{X}$, the identification of matrix $\mathbf{M}$ and the derivation of vectors $\mathbf{p}$ and $\mathbf{s}$ is done offline only once, after which they can be used in any anchor shape identification problem involving four anchors, as we detail in the next subsections.

To the best of our knowledge, this is the first time that the anchor estimation problem is tackled using automatic expression manipulation to estimate anchor shapes as defined above. This approach enables the fusion of ADoA measurements not just for locating a moving client, but also to estimate the location of anchor nodes in a computationally efficient way.

B. Erroneous ADoA measurements

The formulation above assumes error-free ADoA measurements. To take into account measurement errors, assume that N_T different measurements are obtained at different time epochs, indexed by t. For this set of measurements, ideally, the anchor shape would be the one that generates $\mathbf{s}$ such that $\mathbf{p}_t^T\mathbf{s}=0\ \forall t$, where we remark that the anchor angle terms $\mathbf{s}$ do not depend on t. With erroneous measurements, we cannot achieve the equality, hence we resort to a minimum mean square error (MMSE) approach, by defining the cost function

$$\mathcal{F}(\mathcal{S}) = \sum_t(\mathbf{p}_t^T\mathbf{s})^2 = \mathbf{s}^T\left(\sum_{t=1}^{N_T}\mathbf{p}_t\mathbf{p}_t^T\right)\mathbf{s} = \mathbf{s}^T\mathbf{O}\mathbf{s}\,, \quad (4)$$

where $\mathbf{O} = \sum_{t=1}^{N_T}\mathbf{p}_t\mathbf{p}_t^T$. The solution to the anchor shape estimation problem is then obtained as

$$\widehat{\mathcal{S}} = \arg\min_{\mathcal{S}}\ \mathcal{F}(\mathcal{S})\,. \quad (5)$$

In the next section we discuss a practical algorithm that computes and refines the anchor shape with any number of anchors, in a way that is robust to mmWave path obstruction.

III. Practical algorithm and SLAM extension

A. Extension to more than four anchors

Starting from the derivation in Section II, we take all the possible subsets of four anchors, compute the corresponding cost function, and finally sum the obtained values into a global cost function as follows. Call $\mathcal{C}_\mathcal{A}^4$ the set of all possible subsets of four anchors. The global cost function is defined as:

$$\mathcal{F}_\mathcal{A}(\mathcal{S}_\mathcal{A}) = \sum_{c\in\mathcal{C}_\mathcal{A}^4}\mathcal{F}_c(\mathcal{S}_c) = \sum_{c\in\mathcal{C}_\mathcal{A}^4}\mathbf{s}_c^T\mathbf{O}_c\mathbf{s}_c\,, \quad (6)$$

where $\mathcal{F}_c, \mathcal{S}_c, \mathbf{s}_c, \mathbf{O}_c$ are the cost function, the anchor shape, the vector $\mathbf{s}$ and the objective matrix $\mathbf{O}$ considered in Section II, computed for the set of anchors $c\in\mathcal{C}_\mathcal{A}^4$.

We now extend the formulation to account for mmWave path blockage. Note that this causes two separate issues: i) matrix $\mathbf{O}_c$ could be impossible to compute since some anchors in c may not be visible; ii) some anchor locations may be impossible to estimate, e.g., because they have not been observed in a sufficient number of measurements. Define $\mathcal{A}_v^t \subset \mathcal{A}$ as the set of visible anchors at time t. Issue i) is solved by considering only the measurement indices t for which all anchors of c are visible. To achieve this, we consider an update strategy for matrix $\mathbf{O}_c$. As the algorithm assumes no initial knowledge, at $t=0$ we have $\mathbf{O}_c = \mathbf{0}$. At time t, only the anchor sets $c \subset \mathcal{A}_v^t$ permit to update $\mathbf{O}_c$. Call $\mathcal{C}^4_{\mathcal{A}_v^t}$, the set of all subsets of four anchors among those that are visible at time t. For $\mathcal{T}_c^t = \{1 \leq \tau \leq t \text{ s.t. } c \in \mathcal{C}^4_{\mathcal{A}_v^\tau}\}$, we compute

$$\mathbf{O}_c = \sum_{\tau \in \mathcal{T}_c^t} \mathbf{p}_\tau^c \mathbf{p}_\tau^{cT}\,, \tag{7}$$

where $\mathbf{p}_\tau^c$ is the vector $\mathbf{p}$ considered in Section II, computed for set $c \in \mathcal{C}^4_{\mathcal{A}}$ at time τ. Eq. (7) allows us to include the contribution of all valid measurements by adding one more term to each sum whenever a measurement is taken (i.e., t increases by 1).

To solve issue ii), we start by estimating the anchor shape for subsets of anchors that have been observed in a sufficient number of measurements. Then, we iteratively include additional anchors until there are no more anchors that can be accurately located. To do this, we design a set of criteria to compute a first estimation of four anchors, add one additional anchor, and refine the estimation over the new set.

Having the update algorithm for the $\mathbf{O}_c$ matrices, and starting from $\mathcal{A} = \varnothing$ at $t=0$, we pick the first set of four anchors $\mathcal{A}^* \in \mathcal{C}^4_{\mathcal{A}}$ to be the first combination such that all anchors in the set appear at least in θ_I measurements. θ_I is called the "initialization threshold." The initial estimation of the anchor locations is then computed as $\hat{\mathcal{A}}^* = \arg\min_{\mathcal{A}^*} \mathcal{F}_c(\mathcal{S}_c)$. We achieve this via a Nelder-Mead simplex direct search [17] using as an initial point the minimum cost element of a set of several anchor locations configurations[2] drawn independently at random according to a Gaussian $\mathcal{N}(0,1)$ distribution. We remark that the mean and variance of the distribution are not relevant, since the problem is invariant to translation and scaling. To avoid accuracy issues, after every estimation of $\mathcal{A}^*$, we will normalize it to have mean 0 and variance $|\mathcal{A}^*|^{-1}$.

We now extend set $\mathcal{A}^*$ by adding anchors from $\mathcal{A} \setminus \mathcal{A}^*$, provided that each anchor has appeared in at least θ_E measurements along with at least three anchors already in $\mathcal{A}^*$. θ_E is called the "extension threshold" and it allows the position of the new anchor to be accurately estimated. Call the inserted anchor α. Its location is estimated by computing $\hat{\mathcal{A}}^* = \arg\min_\alpha \mathcal{F}_{\mathcal{A}^*}(\mathcal{S}_{\mathcal{A}^*})$ through a very fast random search in $[-2,2]\times[-2,2]$, followed by a Nelder-Mead simplex direct search. The complexity of this step is even lower than that of the initial search, and can be further limited by restricting the computation of $\mathcal{F}_{\mathcal{A}^*}(\mathcal{S}_{\mathcal{A}^*})$ to the θ_{R_E} anchor combinations $c \in \mathcal{C}^4_{\mathcal{A}^*}$ that appear the most in all measurements (where the subscript E refers to the extension step), and by capping the Nelder-Mead running time to θ_{T_E} milliseconds.

All location estimates for all anchors in set $\mathcal{A}^*$ (including the newly added anchor) are then refined by solving

$$\widehat{\mathcal{A}}^* = \arg\min_{\mathcal{A}^*} \mathcal{F}_{\mathcal{A}^*}(\mathcal{S}_{\mathcal{A}^*}) = \arg\min_{\mathcal{A}^*} \sum_{c \in \mathcal{C}^4_{\mathcal{A}^*}} \mathcal{F}_c(\mathcal{S}_c)\,. \tag{8}$$

As before, we restrict the above sum to the θ_{R_R} combinations of anchors that appear the most in $c \in \mathcal{C}^4_{\mathcal{A}^*}$, and cap the running time of the Nelder-Mead method to θ_{T_R} milliseconds.

B. User localization

Once the anchor locations have been estimated, we can localize the user. We first assume that ADoA measurements are error-free. Localizing the user means therefore to find a location compatible with the ADoAs observed from those physical and virtual anchors whose positions have been estimated. We take anchor a_i as a reference, and use anchor a_j to determine the locus of points that generate the angle $\widehat{a_i u a_j}$. Call $\mathbf{a}_i$ and $\mathbf{a}_j$ the coordinate vectors of a_i and a_j, respectively. The above locus is, by definition, the arc of circumference that contains both a_i and a_j, centered at $\mathbf{R}_{\pi/2-\widehat{a_i u a_j}}(\mathbf{a}_j - \mathbf{a}_i)/\sin(\widehat{a_i u a_j}) + \mathbf{a}_i$, where $\mathbf{R}_\beta(x)$ is the counter-clockwise rotation of $\mathbf{x}$ by an angle β. Anchor $\mathbf{a}_i$ is contained in this circumference. The inversion of this circumference to the point $\mathbf{a}_i$ is the line defined by the points $\mathbf{x}$ such that $\langle \mathbf{x} - \mathbf{a}_i\,, \mathbf{R}_{\pi/2-\widehat{a_i u a_j}}(\mathbf{a}_j - \mathbf{a}_i)\rangle = 2\sin(\widehat{a_i u a_j})$. Denote by $\mathbf{u}^{-\mathbf{a}_i}$ the inversion of $\mathbf{u}$ to point $\mathbf{a}_i$. Combining the corresponding expressions for different indices j yields a system of equations of the form

$$\mathbf{Z}_i(\mathbf{u}^{-\mathbf{a}_i} - \mathbf{a}_i) = \mathbf{Y}_i. \tag{9}$$

In order to account for noisy measurements, we estimate $\mathbf{u}^{-\mathbf{a}_i} - \mathbf{a}_i$ by solving the MMSE problem

$$\widehat{\mathbf{q}} = \arg\min_{\mathbf{q}} \|\mathbf{Z}_i\mathbf{q} - \mathbf{Y}_i\|^2 = (\mathbf{Z}_i^T\mathbf{Z}_i)^{-1}\mathbf{Z}_i^T\mathbf{Y}_i\,. \tag{10}$$

By solving $\widehat{\mathbf{q}} = \mathbf{u}^{-\mathbf{a}_i} - \mathbf{a}_i$ for point $\mathbf{u}$, we get the estimate

$$\widehat{\mathbf{u}}_i = \frac{(\mathbf{Z}_i^T\mathbf{Z}_i)^{-1}\mathbf{Z}_i^T\mathbf{Y}_i}{\left\|(\mathbf{Z}_i^T\mathbf{Z}_i)^{-1}\mathbf{Z}_i^T\mathbf{Y}_i\right\|^2} + \mathbf{a}_i\,. \tag{11}$$

Finally, we compute $\widehat{\mathbf{u}}_i$ for all reference anchors a_i and average the corresponding estimates to yield $\widehat{\mathbf{u}} = \mathbb{E}_i(\widehat{\mathbf{u}}_i)$.

C. Spurious estimate filtering

As a final post-processing step, we rule out location estimates affected by a very large error. As such spurious estimates are normally far from the trajectory of a mobile node, we define for each estimate a value $v_t = \min\{\|\hat{\mathbf{u}}_t - \hat{\mathbf{u}}_{t-1}\|, \|\hat{\mathbf{u}}_t - \hat{\mathbf{u}}_{t+1})\|\}$, where t is a position index, and $\hat{\mathbf{u}}_{t-1}$, $\hat{\mathbf{u}}_t$ and $\hat{\mathbf{u}}_{t+1}$ are estimates computed as in Section III-B. If $\hat{\mathbf{u}}_t$ is spurious, v_t becomes detectably large. We therefore discard an estimate t' if $v_{t'} > 2\,\mathrm{med}(\{v\})$, where $\mathrm{med}(\cdot)$ is the median and $\{v\}$ collects all v_t computed as above.

[2]Note that the initial search involves the computation of only one term of the sum in (7), and is run only once. It therefore represents a very low startup cost, for which a resolution of up to 2^{20} elements could be easily afforded.

D. Simultaneous Localization and Mapping (SLAM)

We finally extend the standard physical/virtual anchor matching method to estimate the environment boundaries by leveraging LoS and NLoS paths. Recall that the location of the virtual source of NLoS mmWave paths can be obtained by mirroring the location of the physical AP through the boundary that reflects the NLoS path. Since CLAM estimates the location of both physical and virtual anchors alike, such estimates can be matched based on the AP's ID and exploited to infer the location of environment boundaries. To this end, we first define a criterion to decide whether a wall actually "exists." Assume a virtual anchor $\mathbf{a}_v$ is obtained by mirroring the physical anchor $\mathbf{a}_p$ through a wall. The normal vector pointing towards the inside of this wall is computed as $\mathbf{n}_{pv} = (\mathbf{a}_v - \mathbf{a}_p)/\|\mathbf{a}_v - \mathbf{a}_p\|$. For every physical anchor, we can compute all the $\mathbf{n}_{pv}$ corresponding to its virtual anchors. To avoid mistaking proper virtual anchors for spurious reflections, e.g., caused by scattering, we impose that normal vectors to the same wall computed from different positions must agree. Formally, we check that

$$\sum_{\mathbf{n}'_{pv}} \exp\left(-\frac{\|\mathbf{n}_{pv} - \mathbf{n}'_{pv}\|}{2\beta^2}\right) > \theta_N \,, \tag{12}$$

where $\mathbf{n}'_{pv}$ spans the set of all virtual anchors corresponding to any physical anchor, $\beta^2 = 0.01$ is the variance of the Gaussian kernel and θ_N is a reliability threshold. If (12) is satisfied, we compute a wall position estimate corresponding to the reflection point of a received NLoS path. Such position is obtained as the intersection between the line that bisects segment $\overline{\mathbf{a}_p\mathbf{a}_v}$ and the segment $\overline{\widehat{\mathbf{u}}\mathbf{a}_v}$.

Throughout our simulation campaign, we found $\theta_N = 2.5$ to yield a good tradeoff between the accuracy of the wall estimates and the number of discarded estimates (both increase with increasing θ_N). Thanks to the flexibility offered by the thresholds employed in CLAM, it is possible to adapt to different complexity requirements. For example, users with more powerful devices could increase θ_I, θ_E, θ_R and θ_{R_R} to yield a more accurate solution at the cost of higher complexity. Conversely, power saving or time-constrained operations are enabled by smaller threshold values, or by constraining the optimization problem to a shorter run time via θ_{T_E} and θ_{T_R}. In an extreme case, we can also decide to refresh the shape estimates only once every given number of measurements, so as to avoid continuous computation on constrained devices.

IV. Experimental validation

Currently, there exists no COTS equipment providing direct access to RF chain output, and phased arrays are not easily available for integration in a testbed. Therefore, to evaluate our algorithm in a real environment, we use a Pasternack VubIQ 60-GHz down-converter with a 7°-aperture horn antenna as user device. To emulate sector-level sweeping, we mount the VubIQ on a stepper motor controlled by an Arduino board. The baseband signal is recorded by an Agilent EXA N9010A Signal Analyzer. The setup is controlled by Matlab code running on a laptop. The system is configured to take fine grained mmWave power measurements in the 60 GHz band in angular steps of 0.45°, so that 800 measurements cover a complete circumference. For the APs we employ five transmitters with 60-GHz up-converters: one Pasternack VubIQ, two SiversIMA DC1005V/00 and two SiversIMA CO2201A.

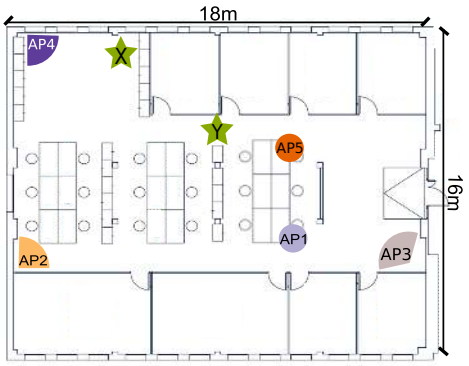
We use pseudo omni-directional transmitters in order to speed up the collection of the measurements, and capture all available paths of a given AP at the same time. As a general rule-of-thumb, we equipped APs located in open areas with an omni-directional antenna; where omni-directionality is not needed we employed wide-beam antennas (e.g., 80°-aperture horn antennas for APs near corners, and open wave-guide terminations translating into a 120° antenna aperture for APs located near walls).

Once the measurements have been collected at different locations, we retrieve the AoA patterns for each AP. Different peaks in each pattern correspond to the LoS arrival (when available) and to one or more NLoS arrivals.

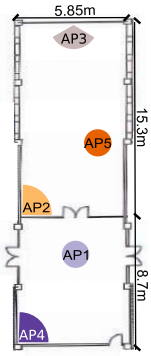
A. Room Setup

We test our algorithms in two different scenarios. The first, Scenario A (Fig. 2a), is an L-shaped working place with an open area, six small offices and two labs, all of which are in active use. The walls are made of bricks and have glass windows. Glass panels divide the offices from the open area. Office and lab furniture (including tables, chairs, screens and metal cupboards) make the propagation setting realistically complex. The omni-directional transmitters were placed centrally in the open area, whereas the 80° horn antennas were positioned in the two west corners of the open area. In this scenario, we measured the AoA pattern from different APs at 66 positions. This makes it possible to emulate different trajectories through the open space, with an average separation of 1.3 m between nearest measurement points. Note that furniture constrained the measurements, which were taken around the work stations, in empty areas, and along the corridors. In Fig. 3 we show two examples of the equipment and furniture in part of scenario A's measurement setup. To provide a good tradeoff between CLAM's accuracy and complexity, we set $\theta_I = 8$, $\theta_E = 5$, $\theta_{R_E} = 64$, $\theta_{R_R} = 128$. We do not set specific values for θ_{T_E} and θ_{T_R}, which provides an upper bound to localization performance. The typical running time is about 500 ms on a mainstream laptop.

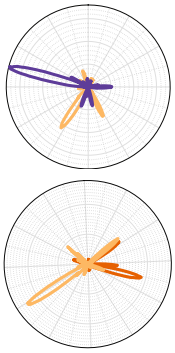
Scenario B (Fig. 2b) consists of a large room of with a size of about 24×6 m^2. The room has brick walls and glass windows, and is mostly empty of furniture. A wooden wall, interrupted by an open door, splits the space into two sections. In the smaller of the two sections we placed the omnidirectional transmitter, slightly shifted from the middle of the room, and deployed the 80°-aperture transmitter in a corner. In the largest space we set up a 80°-aperture transmitter in a corner, and an omnidirectional transmitter slightly south-west from the middle of the room. The 120°-beamwidth transmitter was centrally deployed along the top wall. For this setup, we measured 72 different positions in a scattered grid, where nearest locations are about 1.15 m apart.



(a) Scenario A



(b) Scenario B



(c) Positions X and Y

Fig. 2. Floor plans of Scenarios A and B, showing the locations of mmWave APs and the antenna aperture for each AP (80°, 120° or omni). (c) AoA spectra for APs 2 and 4 at position X and for APs 2 and 5 at position Y. Both LoS and reflected NLoS paths are clearly distinguishable.

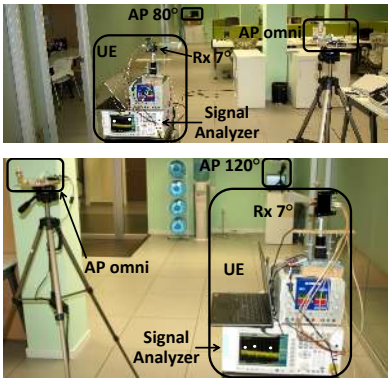


Fig. 3. The measurement setup, showing APs and the receiver. The dots in the analyzer represent signals from different APs.

In Fig. 2c, we depict the received AoA patterns at two positions marked as "X" and "Y" in Fig. 2a. For position X, we show the AoA spectra of APs 4 (purple) and 2 (orange), whereas we consider APs 2 (orange) and 5 (brown) for position Y. As expected, AoA spectra reveal multiple arrivals from different directions, with a LoS arrival of higher power, and weaker NLoS arrivals related to paths that incurred one or two reflections. AoA measurements are fed to the localization algorithm in order to estimate the location of the user based on the procedure in Sections II and III.

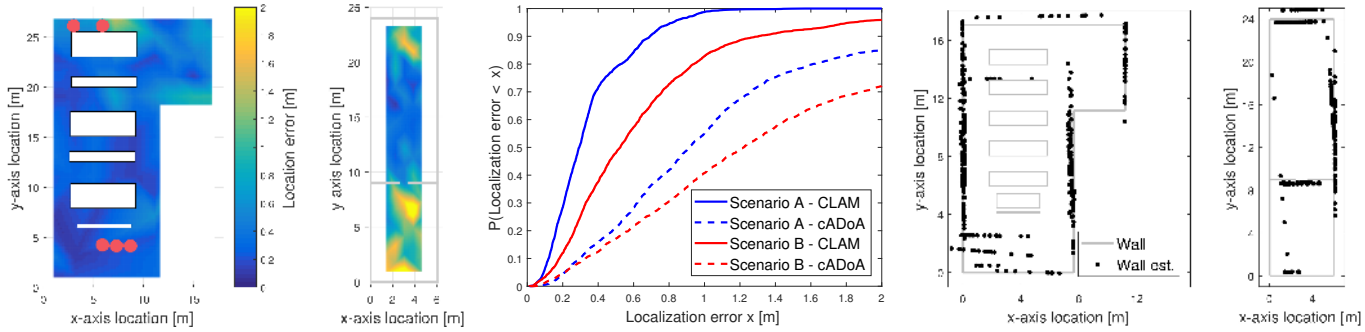
B. Measurements results

In Fig. 4 we illustrate the performance of CLAM over the data collected in our measurement campaign. The heat map in Fig. 4a shows the median user localization error throughout different positions in Scenario A. The voids in the map represent non-walkable locations such as tables, storage, and internal walls. The error is remarkably low (mostly sub-50 cm) everywhere in the room except for a few locations near the top and top-right sections of the area. Here, the larger errors are mostly due to the less accurate positioning of some virtual anchors. In turn, this is due to the weak mmWave illumination that results from the sparse AP deployment and can be reduced significantly by increasing the number of APs. Moreover, some areas are subject to higher errors, and the corresponding location estimates are discarded as outliers (see Section III-C): these areas are marked via a red circle in Fig. 4a. Still, the results are very accurate, given the shape of the room as well as the presence of furniture and other obstacles in the area.

Fig. 4b shows the median localization error of CLAM in Scenario B. We observe that the locations in the top section are accurately estimated, except near the top-right room corner, where the illumination from the omni-directional AP5 (see also Fig. 2b) becomes comparatively weaker. CLAM's accuracy is smaller in the bottom section, where some estimates are affected by a larger error, especially near the door and the bottom wall. These results are expected due to the presence of two APs compared to the three in the top section, which yields poorer illumination from LoS and NLoS arrivals. Besides limited AP coverage, the transition across the middle door in Scenario B causes a disconnect between anchor knowledge on either side. Thus, it is impossible for the mobile node to leverage the three anchors in the top section to keep localizing in the bottom section. As a result, CLAM can only work with two physical anchors and the corresponding virtual anchors. Still, we consider CLAM's accuracy to be remarkable, in particular given that it does not assume any initial knowledge about the user and the environment, and that our deployment is characterized by a low density of APs.

Fig. 4c shows the localization error CDF in Scenarios A and B for both CLAM and the ADoA-based algorithm of [15], named cADoA in the following. Recall that, unlike CLAM, cADoA assumes to know the position of the anchors and the environment map. Despite these assumptions being in favor of cADoA, CLAM achieves a median error smaller than 0.3 m (Scenario A) and 0.5 m (Scenario B), whereas cADoA is affected by a generally larger error, and by a median error about 1 m in scenario A and 1.3 m in scenario B. About 99% of CLAM's location estimates achieve sub-meter accuracy in scenario A. In scenario B, CLAM's accuracy is slightly worse. However, sub-meter errors are still achieved 83% of the time, despite the very challenging case of little AP coverage in the bottom section of the scenario. Location estimation fails (due to insufficient available paths) in 9% of scenario A's positions and in 5% of scenario B's.

We conclude by discussing experimental SLAM results in Fig. 4d for both scenarios A and B. In scenario A, the experimental results show very good estimates of the right and left perimeter walls, which become less accurate for the top and bottom walls. Here, imperfect virtual anchor estimates tend to cause slight mismatches and rotations of wall locations. Overall, the algorithm provides a realistic view of the location of the room walls and obstacles therein, which would be compatible with indoor navigation requirements with no map available a priori. Aggregating these estimates over more measurements (as done in the subsequent simulative analysis) and filtering outliers does allow to build very accurate environment maps. Remarkably, the figure shows that the algorithm correctly estimates some reflective objects in the



(a) Scenario A (b) Scenario B (c) Error CDF (d) SLAM, scenarios A and B

Fig. 4. Experimental results for CLAM: heat maps of the median localization error for (a) scenario A and (b) scenario B; (c) CDF of the localization error. (d) Experimental SLAM results for scenarios A and B: gray lines represent the walls, and each dot a wall location estimate.

open area at the time of the measurement, including screens in the second table from the top.

The SLAM performance is comparatively worse in scenario B, where CLAM correctly estimates the right perimeter wall, together with the top and middle walls, and a few locations in the bottom wall. The left wall is affected by a higher uncertainty, as only a few marked points appear. Still, given the challenging scenario with only a few APs and the fact that environmental sensing comes as an added value of the localization algorithm, we believe these results to be very promising and worth further investigation.

V. Simulation Results

To investigate CLAM's performance in more detail, we now reproduce scenarios A and B of Section IV in simulation. We use a custom ray-tracing based simulator that accurately models the propagation environment, and solves the electromagnetic propagation equations by taking into account the properties of the different materials that constitute the walls and other objects in the scenarios. This allows us to emulate many user trajectories (something that would be too time-consuming in a real measurement), to systematically evaluate mutual mmWave path blockage by the users, and to understand the long-term performance of CLAM when the movement of the user through the environment enables continued refinement of position estimates over time.

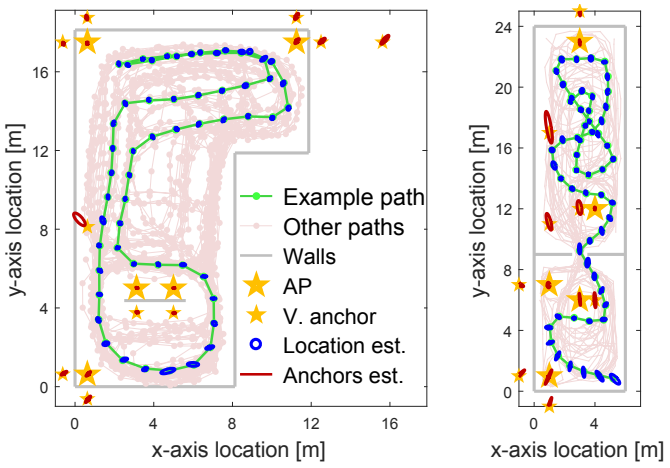
With reference to Figs. 5a and 5b, in each scenario we deploy five physical APs (large stars, where smaller stars represent their virtual counterparts). We consider a total of five users that move along a smooth trajectory at 1 m/s. The trajectories are rendered using light purple lines and points. The green path is one example path we highlight to discuss the localization results below. Each node takes AoA measurements at different locations (marked with dots for the purple paths, or circles for the green path), and applies CLAM to locate the physical and virtual anchor nodes, and to estimate its own position while moving. We simulate the mutual blockage of mmWave paths by modeling the users as circles of diameter 0.6 m, and by assuming that any path crossing such circles does not propagate further. For any given user, between 20% and 25% of the propagation paths are blocked throughout scenario A, and between 15% and 20% in scenario B.

We start with Fig. 6 for scenario A and Fig. 7 for scenario B, which show the cumulative distribution function (CDF) of the user localization error, computed over the ensemble of location estimates taken by all mobile users. CLAM is compared to the two approaches in [15], which belong to the class of triangulation-based and ADoA-based algorithms, respectively named cTV and cADoA. Unlike CLAM, both cTV and cADoA assume to know the position of the anchors and the floor map, so as to be able to locate virtual anchors. All schemes are tested by assuming angle measurements to be affected by a Gaussian error of standard deviation $\sigma = 1^\circ$ and $\sigma = 2^\circ$. These errors correspond to those obtained by synthesizing beam shapes with a uniform linear array with 32 and 16 elements, respectively, under typical SNR conditions.

We observe that CLAM achieves much smaller localization errors than cTV and cADoA, despite starting with zero initial knowledge about the AP locations and the environment. In scenario A, up to 97% of the location estimates are affected by an error of less than 50 cm, with a median error between 10 and 20 cm, depending on the accuracy of the AoA measurements. cADoA and cTV perform worse, especially if AoA data is less accurate ($\sigma = 2^\circ$). CLAM fails to localize a user between 0.5% and 1.5% of the time[3], against 0.06% for cADoA and no failures for cTV. Scenario B (Fig. 7) shows the same trend. However, the inner wall makes the anchor configuration sparser, and increases the probability that users block mmWave paths through the door. This is reflected by CLAM being slightly less accurate than in scenario A. In any event, it still achieves sub-meter accuracy (about 95% of the cases) and a median error of about 15 cm for $\sigma = 1^\circ$. We observe that cADoA and cTV show much larger median and average errors, and a significant probability that the localization error exceeds 2 m for less accurate angle measurements. In scenario B, the percentage of localization failures is between 2% and 4% for CLAM (no failures for cADoA and cTV). We note that if we applied the outlier detection filter to cADoA and cTV, their lack of accuracy would lead to discarding up to 20% of the estimates for scenario A and up to 30% for scenario B.

We now focus on the trajectory of a single user (the green path in Figs. 5a and 5b), in order to investigate the localization

[3]Failure occurs when a point is not illuminated by a sufficient number of anchors, or because an estimate is marked as an outlier as per Section III-C.



(a) Scenario A (b) Scenario B

Fig. 5. Simulations: localization results for scenarios A and B. Localization is quite accurate despite mutual mmWave path blockage among the users.

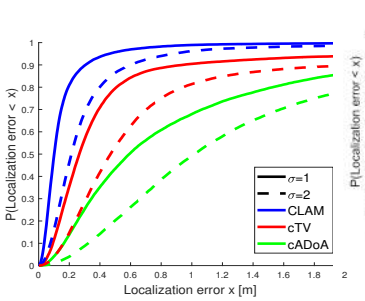


Fig. 6. Simulations: localization error CDF for scenario A.

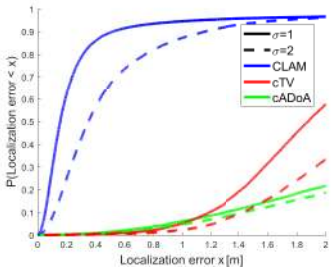
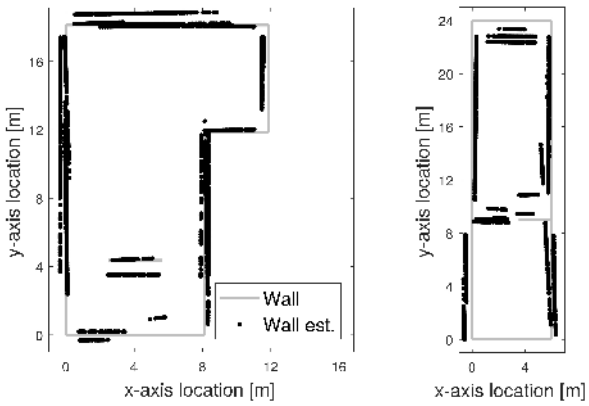


Fig. 7. Simulations: localization error CDF for scenario B.



(a) Scenario A. (b) Scenario B

Fig. 8. SLAM simulation results for (a) scenario A and (b) scenario B. Light gray lines show the walls; each black dot represents a wall location estimate.

errors in more detail. Recall that large stars correspond to physical APs, whereas small stars represent virtual APs, that model the (virtual) sources of NLoS paths. The uncertainty of anchor and user location estimates is rendered through ellipses whose half-axis lengths equate the standard deviation of the estimation error along the corresponding direction. Most of the estimated anchor and user locations are extremely accurate, as seen from the tiny uncertainty contours. This includes all physical and almost all virtual anchors. Just for those anchors that the user observes only from a few positions do the uncertainty ellipses become larger.

As a general observation, if some anchor position estimates are affected by higher uncertainty, the user location estimates relying on those anchors will be less accurate as well. This can be observed in the bottom section of scenario B (Fig. 5b). The frequent mmWave propagation path blockage due to other users limits the number of anchor observations in the bottom section, which translates into lower accuracy. At the same time, users block the visibility of the anchors in either room as they move through the door, which prevents leveraging information from both anchor sets. In the larger section, path blockage is mitigated due to the larger room space, which leads to more accurate location estimates.

We finally show in Fig. 8 the results of the estimation of room walls and obstacles via the SLAM algorithm described in Section III-D. Fig. 8a shows that SLAM works well in scenario A: the location of all walls is estimated with good accuracy, with only a few spurious results. Those are mainly due to the error in the position estimate of some physical and virtual anchors (see Fig. 5a) near the top wall and near the internal wall in the bottom part of the room. The lack of estimates for the bottom right part, instead, is due to an insufficient number of physical-virtual anchor pairs that agree on the existence of a wall, so that (12) does not exceed the threshold θ_N. In scenario B the results are good as well. Given the slightly higher localization errors, a few walls are estimated with the wrong inclination, especially in the bottom section. We remark that SLAM results in both scenarios would become more accurate with a denser AP deployment, or by taking further measurements, e.g., as a user repeatedly walks through the respective indoor areas.

VI. Related Work

The literature most related to our approach lies in the areas of localization and SLAM. We survey each category below.

Localization in mmWave systems— Localization has been largely studied from theoretical and practical standpoints [18]. Localization is considered as an inherent feature of mmWave communications [19], and thanks to the characteristics of mmWave signals is potentially achievable with up to sub-centimeter accuracy. Large-scale mmWave MIMO systems have been leveraged for localization by detecting the changes in the statistics of sparse MIMO channel signatures [20]. From a theoretical standpoint, [21] shows that in a number of practical cases it is possible to estimate both the position and the orientation of a user. This is in line with the capabilities of our proposed algorithm.

The quasi-optical propagation as well as the sparse AoA spectrum perceived by a mmWave receiver enable single-anchor localization, with much better accuracy than can be achieved in microwave systems [22], especially if the environment and the AP locations are assumed unknown. For example, [15] achieves this via two approaches belonging to the class of triangulation- and ADoA-based algorithms. In the same vein, [14] applies ranging and multilateration

to to exploit LoS and NLoS arrivals for node localization. The work in [16] is a first attempt to achieve localization without initial knowledge of the environment. However, no experimental results are provided.

Simultaneous localization and mapping— SLAM is a fundamental problem in the field of computer vision applied to robotics [23]. The original solution employed a laser range scanner (or other range, bearing or odometry sensors) and fused the measurements via an extended Kalman filter (EKF). Laser or lidar scanning and depth cameras have then evolved to GPU-based or cloud implementations [24] that reduce complexity by offloading computationally intensive tasks. A range-based algorithm that exploits accurate range measurements in a multipath environment was proposed for SLAM in [25]. SLAM has also been realized through bearing-only sensors, often in conjunction with the EKF or other robust smoothing techniques, such as Rao-Blackwellized particle filters [26].

Our CLAM algorithm achieves SLAM in mmWave networks through ADoA information extracted from standard-compliant beam training and tracking algorithms. Unlike [14], [27] and to abide to COTS device constraints, we do not perform ranging, and unlike [15], [22] we assume no a-priori information. Our algorithm is distributed and does not require the nodes to cooperate (unlike, e.g., [28]). Anchor locations are estimated by exploiting all possible relationships among the angles measured by a client at different locations. To the best of the authors' knowledge, it is the first time such a formulation is used, enabling a high-accuracy and low-complexity implementation, including fast initialization.

VII. Conclusions

In this paper, we proposed CLAM, a zero-initial information, zero-overhead and low-complexity SLAM approach tailored to the characteristics of mmWave networks. We leverage only standard procedures such as beam-training to retrieve angle-difference-of-arrival information, and use this information to estimate the location of the mmWave APs, the position of the user, and the surrounding environment. Low-complexity estimation procedures are enabled by a fundamental reformulation of ADoA-based anchor location estimation.

We demonstrate that CLAM localizes with very high accuracy, and that it is robust to mmWave signal blockage due to human movement in the localization area. Our results are based on extensive experimental measurements in two different areas, including a fully operational working area, as well as additional simulations. The outcomes validate the accuracy of CLAM and demonstrate that it is a feasible approach for realistic mmWave network deployments.

Acknowledgment

This work has been supported in part by the ERC project SEARCHLIGHT, grant no. 617721, the Ramon y Cajal grant RYC-2012-10788, the grant TEC2014-55713-R (Hyperadapt) and the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919).

References

[1] S. Rangan, T. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014.

[2] Z. Pi, J. Choi, and R. Heath, "Millimeter-wave gigabit broadband evolution toward 5G: fixed access and backhaul," *IEEE Commun. Mag.*, vol. 54, no. 4, pp. 138–144, Apr. 2016.

[3] TP-Link. Talon AD7200 multi-band WiFi router. [Online]. Available: http://www.tp-link.com/us/products/details/cat-5506_AD7200.html

[4] NETGEAR. Nighthawk X10 smart WiFi router. [Online]. Available: https://www.netgear.com/landings/ad7200/

[5] T. Rappaport *et al.*, "Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications," *IEEE Trans. Antennas Propag.*, vol. 61, no. 4, pp. 1850–1859, Apr. 2013.

[6] J. G. Andrews *et al.*, "Modeling and analyzing millimeter wave cellular systems," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 403–430, Jan. 2017.

[7] M. Giordani, M. Mezzavilla, and M. Zorzi, "Initial access in 5G mmWave cellular networks," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 40–47, Nov. 2016.

[8] Y. Kim *et al.*, "Feasibility of mobile cellular communications at millimeter wave frequency," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 589–599, Apr. 2016.

[9] T. Nitsche *et al.*, "IEEE 802.11ad: directional 60 ghz communication for multi-Gigabit-per-second Wi-Fi," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 132–141, Dec. 2014.

[10] M. E. Rasekh *et al.*, "Noncoherent mmWave path tracking," in *Proc. ACM HotMobile*, Sonoma, CA, USA, Feb. 2017.

[11] S. Sur *et al.*, "BeamSpy: Enabling robust 60 GHz links under blockage," in *Proc. USENIX NSDI*, Santa Clara, CA, Mar. 2016.

[12] M. K. Haider and E. W. Knightly, "Mobility resilience and overhead constrained adaptation in directional 60 GHz WLANs: Protocol design and system implementation," in *Proc. ACM MobiHoc*, Paderborn, Germany, Jul. 2016.

[13] M. Simsek *et al.*, "5G-enabled tactile internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.

[14] J. Chen *et al.*, "Pseudo lateration: Millimeter-wave localization using a single RF chain," in *Proc. IEEE WCNC*, San Francisco, CA, Mar. 2017.

[15] A. Olivier *et al.*, "Lightweight indoor localization for 60-GHz millimeter wave systems," in *Proc. IEEE SECON*, London, UK, Jun. 2016.

[16] J. Palacios, P. Casari, and J. Widmer, "JADE: Zero-knowledge device localization and environment mapping for millimeter wave systems," in *Proc. IEEE INFOCOM*, Atlanta, GA, May 2017.

[17] J. C. Lagarias *et al.*, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[18] J. Aspnes *et al.*, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006.

[19] F. Lemic *et al.*, "Localization as a feature of mmWave communication," in *Proc. IWCMC*, Paphos, Cyprus, Sep. 2016.

[20] H. Deng and A. Sayeed, "Mm-wave MIMO channel modeling and user localization using sparse beamspace signatures," in *Proc. IEEE SPAWC*, Toronto, Canada, Jun. 2014.

[21] A. Shahmansoori *et al.*, "5G position and orientation estimation through millimeter wave MIMO," in *Proc. IEEE GlobeCom*, Dec. 2015.

[22] P. Meissner *et al.*, "Accurate and robust indoor localization systems using ultra-wideband signals," in *Proc. ENC*, Vienna, Austria, Apr. 2012.

[23] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[24] S. Dey and A. Mukherjee, "Robotic SLAM: A review from fog computing and mobile edge computing perspective," in *Proc. MOBIQUITOUS*, Hiroshima, Japan, Nov. 2016.

[25] H. Naseri and V. Koivunen, "Cooperative simultaneous localization and mapping by exploiting multipath propagation," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 200–211, Jan. 2017.

[26] M. Deans and M. Hebert, *Experimental comparison of techniques for localization and mapping using a bearing-only sensor*. Springer Berlin Heidelberg, 2001, pp. 395–404.

[27] D. Moore *et al.*, "Robust distributed network localization with noisy range measurements," in *Proc. ACM SenSys*, Baltimore, MD, Nov. 2004.

[28] J. Xu, M. Ma, and C. L. Law, "AoA cooperative position localization," in *Proc. IEEE GLOBECOM*, New Orleans, LA, Nov. 2008.