Position Tracking using Likelihood Modeling of Channel Features with Gaussian Processes

Sebastian Kram*, Christopher Kraus*, Tobias Feigl†, Maximilian Stahlke*, Jörg Robert†, and Christopher Mutschler†

{ sebastian.kram, krauscr, tobias.feigl, maximilian.stahlke, christopher.mutschler }@iis.fraunhofer.de
joerg.robert@tu-ilmenau.de

* Fraunhofer IIS, Fraunhofer Institute for Integrated Circuits IIS, Nuremberg, Germany
† Technical University of Ilmenau, Ilmenau, Germany

Abstract—Recent localization frameworks exploit spatial information of complex channel measurements (CMs) to estimate accurate positions even in multipath propagation scenarios. State-of-the-art CM fingerprinting (FP)-based methods employ convolutional neural networks (CNN) to extract the spatial information. However, they need spatially dense data sets (associated with high acquisition and maintenance efforts) to work well—which is rarely the case in practical applications. If such data is not available (or its quality is low), we cannot compensate the performance degradation of CNN-based FP as they do not provide statistical position estimates, which prevents a fusion with other sources of information on the observation level.

We propose a novel localization framework that adapts well to sparse datasets that only contain CMs of specific areas within the environment with strong multipath propagation. Our framework compresses CMs into informative features to unravel spatial information. It then regresses Gaussian processes (GPs) for each of them, which imply statistical observation models based on distance-dependent covariance kernels. Our framework combines the trained GPs with line-of-sight ranges and a dynamics model in a particle filter. Our measurements show that our approach outperforms state-of-the-art CNN fingerprinting (0.52 m vs. 1.3 m MAE) on spatially sparse data collected in a realistic industrial indoor environment.

Index Terms—Gaussian process regression, autoencoder, channel state information, multipath-assisted position estimation.

I. INTRODUCTION

Classical radio-frequency (RF) positioning mostly relies on the extraction of signal propagation times (or their differences) and a subsequent multi-lateration. While this works well under line-of-sight (LOS) propagation conditions, the performance of these approaches deteriorates heavily in indoor environments where multipath propagation is predominant. This is why approaches for dealing with multipath (MP) propagation have been investigated in recent years [1], [2]. Specifically, the large bandwidths offered by modern RF positioning systems such as ultra-wideband (UWB) allows for an increased temporal (and therefore spatial) resolution of propagation paths [3].

Since MP propagation produces valuable spatial information [4], [5] it is desirable to exploit it for positioning. However, the complete extraction of such spatial information embedded in channel measurement (CM) vectors is still an active research area, as different propagation effects such as reflection, diffraction or scattering contribute different kinds of information, some of which are hard to model analytically. Methods such as channel simultaneous localization and mapping (C-SLAM) [6], [7], error mitigation (EMI) [8]–[10], and fingerprinting (FP) [11]–[14] exploit or mitigate the spatial characteristics of environment-related signal propagation using information from CM. In contrast to C-SLAM and EMI, which both typically rely on observation likelihood modeling, FP formulates a regression or classification task with either the complete CM or extracted features thereof as the input, and the positions as an output of a neural network. Still, unlike C-SLAM and EMI, FP exploits additional, diffuse multipath information and hence, can perform well in densely cluttered environments, see Fig. 1: the LOS component that EMI mainly relies on, and strong MPCs caused by reflecting surfaces that C-SLAM requires, are not as readily available.

However, state-of-the-art FP has a number of downsides. First, as it directly estimates the position it does not model an observation likelihood explicitly. This leaves much of its potential unused as information fusion with other sources of spatial information such as time-of-flight (TOF) is only possible on the state level of the tracking target. Thus, to fuse all these approaches on the observation level, we need to represent spatial information as an observation likelihood model. Second, FP requires large, spatially dense datasets. This is difficult in practice as such datasets must be acquired with expensive positioning reference systems and adapted to environment changes. Third, FP only outperforms EMI in areas with complex signal propagation and low LOS, i.e., in smaller areas within the environment (such as densely packed production spaces). Thus, it is desirable to resort to FP only in parts of the environment to lower the database acquisition and maintenance effort. Fourth, FP requires all CMs at a central point for processing which results in significant communication load. Hence, in practice FP requires compressed CMs to scale to a large number of devices.

To address these issues jointly (i.e., FP observation likelihood modeling, spatially sparse datasets, and compact CM representation), we propose Gaussian process regression (GPR) of features extracted from CM using an autoencoder (AE) neural network [15] and model-based propagation-related features [8], [12], [16] to generate a statistical observation likelihood model. Unlike state-of-the-art FP methods based on deep (convolutional) neural networks (CNNs) [11], [14], [17], GPR requires less labeled data. Furthermore, it estimates an obser-
viation likelihood model based on proximity to FP observations instead of positions directly. Thus, it can be directly integrated into a tracking filter, e.g., a particle filter (PF).

We study different architectures and parameter configurations for both AE and GPR using real-world data obtained in a realistic industrial indoor environment. We benchmark the positioning and sensor fusion capabilities against state-of-the-art EMI and FP approaches in terms of positioning accuracy and data requirements. Our evaluation focuses on a realistic indoor industrial environment with various interfering objects and a low number of only three anchors. Our position estimates are more accurate than those of EMI for a dense FP dataset, while state-of-the-art FP based on a CNN (CNN-FP) only slightly outperforms our model (as expected). However, when considering a spatially sparse FP dataset, unlike CNN-FP, our model estimates its own reliability and therefore exhibits smaller performance decrease, resulting in a significantly more accurate tracking than the baselines. Therefore, accurate tracking is possible with smaller databases that are easier to obtain and maintain in practical applications.

The remainder of the article is structured as follows. Sec. III describes the system model and formulates the problem. Sec. IV discusses related work. Secs. V and VI describe our solution. Sec. VII describes the evaluation setup. Sec. VIII evaluates the proposed feature selection, while Sec. IX discusses experimental results. Sec. X concludes.

II. MULTIPATH-ASSISTED AGENT TRACKING IN UWB CHANNELS

The spatial information contained in CMs can be modelled as follows (see Fig. 1): At timestep $k$, an anchor, located at $s_j$, sends a signal $s(t)$ of pulse length $T_p$ in the UWB baseband to a mobile agent at $x^k$. The agent receives

$$r(t) = s(t) \ast h(t) + w(t),$$

where $s(t) \ast h(t)$ represents the complex spatial component (via a convolution of $s(t)$ with the channel impulse response $h(t)$) and $w(t)$ describes the complex non-spatial noise components, consisting of, e.g., thermal noise or errors introduced by the processing chain. $w(t)$ is typically modelled as a temporally uncorrelated white Gaussian noise $w(t) \sim \mathcal{CN}(0, \sigma_w^2)$ of the double-sided power-spectral density of $N_0/2$.

$h(t)$ contains both spatially deterministic and diffuse components. The first corresponds to distinguishable propagation paths. The latter contains unsolvable multipath components (MPCs). Depending on the existence $e_0 \in \{0, 1\}$ of the LOS component, each of the $M$ or $M+1$ propagation paths can be defined by a delay $\tau_{m'}$ and corresponding signal attenuation $\alpha_{m'}$. The diffuse noise component $\nu(t)$ contains additional MPCs caused by the environment through diffraction, rough scatterers or higher order reflections. This process is usually modeled as a zero-mean random process of the time-dependent power delay profile $S_\nu(\tau) \delta(\tau - u) = \mathbb{E}[\nu(t) \ast \nu(u)\ast]$ [5]:

$$h(t) = \epsilon_0\alpha_0\delta(t - \tau_0) + \sum_{m=1}^{M} \alpha_m\delta(t - \tau_m) + \nu(t).$$

To employ this information to track mobile agents, at each timestep $k$, a set of CMs $\mathcal{R}_k = \{ r_k^1(t), ..., r_k^J(t), ... \}$ is available from anchors located at $x_j$, each consisting of a complex time series of length $L$, that can alternatively be interpreted as a $L \times 1$ complex vector $r_k^j$. The goal of a tracking algorithm is then to use subsequent sets of CMs and initial agent position $x_0^k$ to track the agent position $x^k$:

$$x^k = \arg\max_{x^k} p(x^k | \mathcal{R}_k, \mathcal{R}_k^{-1}, ..., x^0).$$

Classical approaches [18] interpret the maximum of the CM as corresponding to the direct LOS connection. Thus, each $r_k^j \in \mathcal{R}_k$ is compressed into a single value, i.e., a noisy estimate of $d_k^j = ||x^k - s_j||_2$ or a similar quantity. However, not only is there much more information contained in the channel as introduced with (2), in indoor environments the LOS assumption is also often violated [12], [19]. It is therefore desirable to exploit more spatial information from $\mathcal{R}_k$ if necessary, while including (applicable) LOS information.

III. RELATED WORK

A variety of different methods for exploiting additional information in CMs have been published. For instance, error mitigation (EMI) approaches rely on the correction of ranges by estimating systematic offset statistics with regression [8], [9] or classify adverse channel conditions [20], [21]. Hence, they produce a correction indicator that can be used for the enhancement of LOS range estimates by offset mitigation, error statistic estimation, or the exclusion of unreliable LOS ranges. Although EMI leads to a significant performance accuracy gain over classical methods, it only focuses on the LOS component. Hence, it requires anchor redundancy and does not consider additional spatial components as a separate source of spatial information.

To circumvent these limitations, C-SLAM [6], [7], [23], [24] relies on exploiting MPCs rather than mitigating their effect on range estimation. Each CM is first decomposed into a set of MPC delays (or, additionally amplitudes [25]), each of which is then associated with a specific reflecting surface or strong scatterer within the environment to produce additional delay measurements originating from a corresponding virtual

Fig. 1: Propagation conditions in empty (left) and cluttered (middle/right) rooms: In empty rooms we see clear LOS connections (green straight line) between $x^k$ and $s_j$ but walls cause distinguishable, deterministic specular reflections (blue dashed lines). When multiple objects are introduced to the environment, these deterministic propagation components are shadowed (red clouds, center), while the objects themselves cause diffuse interaction like diffraction and scattering (purple dotted line, right).

To circumvent these limitations, C-SLAM [6], [7], [23], [24] relies on exploiting MPCs rather than mitigating their effect on range estimation. Each CM is first decomposed into a set of MPC delays (or, additionally amplitudes [25]), each of which is then associated with a specific reflecting surface or strong scatterer within the environment to produce additional delay measurements originating from a corresponding virtual
an additional information extraction layer (see Sec. IV), evaluating different feature combinations to represent the spatial information contained in CMs. GPR has also been used in relation to CMs [36] to estimate deterministic MPCs for CS-LAM. Instead, we employ GPR as a tool for observation modeling in FP (see Sec. V). Our method differs from the state-of-the-art in CM-assisted positioning. First, unlike EMI and LOS approaches, we use additional spatial information contained in MPCs for tracking. Therefore, we can track with lower numbers of anchors and in environments without LOS. Second, unlike CNN-FP, we use a compact feature-space representation of the CM, reducing communication bandwidth requirements. Furthermore, as we use GPR we learn observation likelihood models that also estimate the information reliability. We can thus combine the information on an observation level, providing reliability estimates for all information sources and so, do not require spatially complete data.

IV. INFORMATION EXTRACTION

We extract spatially significant signal features from each complex-valued CM vector $r \in \mathbb{C}^L$, resulting in a low-dimensional representation $f \in \mathbb{R}^F$, see Information Extraction in Fig. 2. We omit anchor and time indices for clarity (e.g., $r \triangleq r_j$) if it is clear from the context.

A. LOS Extraction

First, we classify if $r$ contains a LOS component, using a state-of-the-art LOS identification algorithm [9]. Hereby, we calculate a LOS quality parameter $\beta_0$ using an anomaly detection algorithm, trained on a set of unlabeled and anchor- and environment-independent CMs obtained in a pure LOS environment. Therefore, the algorithm provides an estimate $\hat{\epsilon}_0 \in \{0, 1\}$ of the LOS existence $\epsilon_0$. If $\beta_0$ is above a threshold, we assume that $\epsilon_0 = 1$. In this case, we obtain and evaluate a distance estimate $d$.

B. Feature Extraction

We select well-known propagation-inspired features: the signal energy (ENG), signal decay times (SDT50/SDT75), minimum delay index (MDI), root-mean-square delay spread (RMSDS), Ricean k-factor (RKF), skewness (SKE), and kurtosis (KUR). For details on their computation and additional
features we refer the interested reader to [8], [12], [16]. We compare the suitability of the selected features in Sec. VII-B.

However, previous work [15] has shown that the latent space representation of CMs learned via autoencoders (AEs) compactly represents the information contained in CMs. Hence, we investigate AE-based feature extraction for our positioning task. We train the encoder and the decoder by processing the CM magnitudes $|r|$. The AE implicitly yields a latent space representation $\alpha \in \mathbb{R}^A$. As AEs work unsupervised, we train our AE on an unlabeled set of training data obtained in the same environment. Hence, the AE is trained to compress the information from CM without considering the observation position, i.e., there is no positional bias. As it is unclear which (non-)spatial signal-components are encoded in the latent space and which of them are beneficial for tracking, we analyze different subsets of the elements of $\alpha$ in Sec. VII.

By combining the extracted features, we obtain a feature vector $f \in \mathbb{R}^F$ containing elements of the propagation $p$ and/or AE features $a$ in addition to the LOS distance $d$ if $\hat{e}_0 = 1$. This is depicted in Fig. 2.

V. INFORMATION FUSION

To exploit the extracted features $f$ for positioning, however, they have to be combined with previously collected observations (i.e., FPs) as a straightforward geometrical modeling is impossible. Specifically, in this case a sparse spatial representation is of interest: In cases where a LOS connection to anchors is available, $\hat{d}_x^k$ provides enough spatial information for accurate tracking. So, there is no need to assist tracking with recorded observations. However, in cases where (especially diffuse) MPCs dominate propagation, the LOS component is either not available or superimposed by MPCs, so that $f$ has to provide additional spatial information to maintain accurate tracking.

A. Tracking Problem

Based on consecutive feature and range estimates extracted from sets of observed CMs $\mathcal{R}^k$, we describe our estimation problem of tracking the agent state $x^k$. An FP database $\mathcal{Y}$ and an a-priori state estimate $p(x^k|x^{k-1})$ provide additional spatial information:

$$p(x^k|\mathcal{R}^k, \mathcal{Y}) \propto p(\mathcal{R}^k|x^k, \mathcal{Y}) p(x^k|x^{k-1}).$$

(4)

Using the extracted feature vectors $f^k_j$ (Sec. IV), the observation likelihood $p(\mathcal{R}^k|x^k, \mathcal{Y})$ can be reformulated as a combination of the likelihoods of the extracted LOS ranges and features. While the observation assumption of the LOS features depends on $\hat{e}_0$, the observation likelihood is given by a simple geometric model and additive Gaussian noise with standard deviation $\sigma_{d}$, which is either available from device documentation or can be estimated. Hence, $p(d^k_j|x^k) \sim \mathcal{N}((|x^k - s_j|), \sigma_d^2)$. The feature observation likelihoods $p(z^k_{j,f}|x^k, \mathcal{Z}^{j,f})$ are inferred by training data $\mathcal{Z}^{j,f}$ for each feature/anchor combination, so that we factorize $p(\mathcal{R}^k|x^k, \mathcal{Y})$ from (4) into

$$p(\mathcal{R}^k|x^k, \mathcal{Y}) \propto \prod_{j=1}^J p(d^k_j|x^k) \propto \prod_{j=1}^J \prod_{f=1}^F p(z^k_{j,f}|x^k, \mathcal{Z}^{j,f})$$

(5)

to separate the contributions from each feature. Note that hereby we assume statistical independence between the features.

B. Feature Observation Likelihood Representation

Unlike it is the case for $d^k_j$, for the extracted feature vector a straightforward geometrical derivation of the observation likelihood is not possible. It contains information on environment-specific propagation conditions, both spatially deterministic and diffuse. Therefore, we apply GPR [30] to estimate the observation likelihood from our FP database $\mathcal{Y}$. We train a GPR model $g_{j,f}$ for each feature/anchor combination using $Z_{f,j}$ training data $\mathcal{Z}^{j,f} = \{(z^1_{j,f}, x^1_{j,f}), ..., (z^N_{j,f}, x^N_{j,f})\}$, consisting of positions $x_{j,f}$ and feature observations $z_{j,f}$. We fit an individual standard scaler on all $\mathcal{Z}^{j,f}$ for each feature/anchor combination for numerical stability and apply it to all FPs and test data.

The GPR models the statistics of observed data by fitting covariance kernels $k_{j,f}(x_1, x_2)$. They model the statistical dependence between observations based on their positions in space. If the process is intrinsically stationary, the covariance between observations only depends on the Euclidean distance between them. Thus, it can be modeled using a distance-dependent stationary kernel function $k_{j,f}(||x_1 - x_2||)$. $k_{j,f}$ is obtained by fitting a set of hyperparameters to the training data. As we apply GPR to the extracted features, a straightforward derivation of a suitable kernel function is impossible. Hence, we evaluate different kernel function used in related work in Sec. VII. Each trained GP then regresses the observation likelihood function $p(z^k_{j,f}|x^k, \mathcal{Z}^{j,f})$, represented by a Gaussian distribution:

$$p(z^k_{j,f}|x^k, \mathcal{Z}^{j,f}) \approx g_{j,f}(x^k) \sim \mathcal{N}(\mu_{f,j}(x^k), \sigma^2_{f,j}(x^k)).$$

(6)

The kernel depends on the distance between $x^k$ and each previous observation position $x_{j,f}$. Hence, $\sigma_{f,j}$ significantly increases in areas with close proximity, while $\mu_{f,j}$ converges to the global mean. This is depicted in Fig. 3. Each subfigure visualizes $g_{j,f}(x^k)$ for a single, arbitrary feature value. On the left side, $\mathcal{Z}^{j,f}$ includes recordings distributed over most of the environment. Therefore, $\sigma_{f,j}$ is low in most of the area, only increasing toward the outer edges of the environment. On the right side, we see a distribution based on a smaller, sparse set of data, which, for the application case could be consisting of only data in close proximity to the objects, where diffuse multipath is dominant. Consequently, $\sigma_{f,j}$ increases rapidly (red color) outside of these areas, indicating a decrease in confidence that can be reflected in the weighting of the information source. This is beneficial for positioning, as in these areas, without interfering
objects, the presence of a usable LOS component is more likely, so that $\hat{d}_j$ can serve as a primary source of spatial information.

C. Tracking Filter

To now solve the tracking problem formulated in .(4), we set up a PF [37, 38], visualized on the right side of Fig.2. We represent $p(\mathbf{x}^k)$ by $P$ agent particles $X^k_p \in \mathcal{X}^k; p \in \{1, \ldots, P\}$ with an agent state vector of position and velocity hypothesis and weights, i.e., $X^k_p = \{x^k_p, \mu^k_p, \sigma^k_p\}$. In the prediction step, we propagate the set of particles $\mathcal{X}^k$ using constant velocity model fitted to the agent state dynamics to realize $p(\mathbf{x}^{k|k-1})$, yielding the set of predicted particles $\mathcal{X}^{k|k-1}$.

For the update step, we first use our information extraction pipeline from Sec. IV to obtain $\hat{e}^k_{0,j}$, $\hat{d}^k_j$, and $f^k_j$. Next, we obtain the weights of each individual particle by evaluating our models for all observation log-likelihoods: In the LOS case ($\hat{e}_0 = 1$), the weight $w^k_{j,\text{LOS},p}$ of the obtained TOF is given by the model for $p(d^k_j|x^k)$:

$$w^k_{j,\text{LOS},p} = \frac{1}{\hat{e}_0} \exp\left(-\frac{1}{2} \left(\frac{d^k_j - |x^k_p| - s_j|}{2\sigma^2_d}\right)^2 \right)$$

Hence, for the NLOS case ($\hat{e}_0 = 0$) we assign uniform weights. Then, for all selected features, the corresponding weight $w^k_{j,f,p}$ is given by the observation log-likelihood of the GP modeling $p(z^k_j|x^k, \mathcal{Z}^j)$:

$$w^k_{j,f,p} = \frac{1}{\sqrt{2\pi\sigma^2_{j,f}}} \exp\left(-\frac{1}{2} \left(\frac{z^k_j - \mu^k_{j,f}(x^k|k-1)}{\sigma^k_{j,f}(x^k|k-1)}\right)^2 \right)$$

For each anchor, we assign $W^k_j = [w^k_{j,1}, \ldots, w^k_{j,F}]$. We then normalize all weights for numerical stability so that $\sum_{p=1}^P w^k_{j,f,p} = 1\forall j, f$. Finally, we sum up all the obtained weights to model the combined likelihood $p(\mathcal{R}^k|x^k, \mathcal{Y})$:

$$w^k_p = \sum_{j=1}^J \sum_{f=1}^F \hat{e}_0 w^k_{j,\text{LOS},p} + \sum_{f=1}^F w^k_{j,f,p}$$
kernels, all of which are commonly used stationary kernels [10]. In a preliminary study we obtained the best results with Matern52, which relies on modified Bessel functions that fit well to 2D wave propagation. Hence, we only show results with this kernel. We split data into training (90%) and validation (10%) sets and train the GPs for 500 iterations, storing the model with the lowest sum of log-likelihood distances (see [9]) on the validation data.

**AE:** We investigate three different AE architectures, each with \(4, 6, 8\) latent variables. We train them on unlabeled CM magnitudes \(|r|\) from the dataset. Hereby, we use CMs from more anchors resulting in approx. 37,000 datapoints, split into 80% training and 20% validation data. Our first architecture (FCN-AE) employs a fully connected AE with two hidden layers of size 150 and 80 per encoder and decoder and a rectified linear unit (ReLU) activation function. Our second architecture (CNN-AE-S) uses the best-performing AE from [15] with three convolutional layers per encoder/decoder with exponential linear unit (ELU) and ReLU activation functions. Our third architecture (CNN-AE-P) is a modification of CNN-AE-S that employs the same layer sizes throughout the whole model, such that we apply a max pooling of size 2 after the activation function of each convolutional layer in the encoder. This halves the layer size. In the decoder, we replace the convolutional layers with transposed convolutional layers (up-convolutions), which symmetrically upsample and double the layer size. The idea is to reduce the number of parameters and thus, to improve generalization and to increase the receptive field of the convolutional layers.

**PF:** We evaluate the PF derived in Sec. V.C with \(P = 10,000\) particles and repeat all experiments twice for feature selection and 200 times for the final positioning results to assure stability of our results. We initialize the original state estimate by sampling from \(\mathcal{N}(I_2)\). We initialize all \(x^0_j\) by sampling from \(\mathcal{N}(x_{0,\text{ref}}, 1\mu m^2 I_2)\), where \(x_{0,\text{ref}}\) is the initial position in the reference data. For \(v^0_j\), we initialize the \(x\) and \(y\) components by sampling from \(\mathcal{N}(\mu_v, \sigma^2_v)\), where \(\mu_v\) and \(\sigma_v\) are obtained by taking the absolute mean and standard deviation of the velocities in our training data.

**C. Baseline Methods**

We compare our tracking method against state-of-the-art, we compare the results with both EMI and CNN-FP approaches:

**EMI.** We first detect NLOS [9] to exclude unreliable TOFs using a LOS quality indicator threshold (\(\beta \) of 1.5) obtained by analysis of the data. We combine the TOFs classified as LOS in a range-based PF. For a fair comparison, we implement a dynamic model to assure a fair comparison. A subsequent PF is not applicable in this case, as the trained CNN only produces a position estimate from the recorded data directly.

**C. Baseline Methods**

We compare our tracking method against state-of-the-art, we compare the results with both EMI and CNN-FP approaches:

**EMI.** We first detect NLOS [9] to exclude unreliable TOFs using a LOS quality indicator threshold (\(\beta \) of 1.5) obtained by analysis of the data. We combine the TOFs classified as LOS in a range-based PF. For a fair comparison, we implement a dynamic model to assure a fair comparison. A subsequent PF is not applicable in this case, as the trained CNN only produces a position estimate from the recorded data directly.

**CNN-FP.** We train a CNN-based position regressor [11], on the same data we use to train our GPR models. We use a 6 layer CNN architecture that consists of 4 convolutional layers for feature extraction and 4 fully connected layers for position estimation. The input tensor consists of the recorded complex CMs shifted by the estimated TOFs. We use ReLU activation for all layers, except for the last layer that uses a linear activation. We omit local pooling as this may harm the accuracy on time-series tasks [43]. Nevertheless, to enhance the receptive field of the CNN we enlarge the kernel sizes with the depth of the model: The first convolutional layer uses a kernel size of \([1 \times 3]\), the second \([2 \times 5]\), the third \([3 \times 15]\), and the last \([3 \times 30]\). As CNNs do not include dynamic information, we smooth the CNN position estimates using a linear Kalman filter (LKF) with a constant velocity dynamic model to assure a fair comparison. A subsequent PF is not applicable in this case, as the trained CNN only produces a position estimate from the recorded data directly.

**VII. FEATURE ANALYSIS**

As mentioned in Sec. IV, the effects of the different extracted features on positioning accuracy are unclear in many cases. Also, unlike for the TOFs and similar measures [44], to our knowledge no theoretical positioning error bounds for them have been deduced.

**A. Spatial Analysis**

To interpret the spatial behavior of the GPs, we conduct a spatial analysis of a representative selection. Note that all GPRs work on scaled data, so that the visualization is in arbitrary units. First, we consider the magnitude kurtosis (KUR) (see Figs. 5a and 5b). For the full dataset (Figs. 5a), GPR learns a detailed spatial distribution for \(\mu_{j,f}\) with distinct areas of high (red/orange color) and low (green color) values. Outside of the recording area, \(\mu_{j,f}\) converges to a global mean of 0 due to the scaling. Based on this distribution different areas can be distinguished from each other, e.g., the area behind the absorber wall around \([x, y] = [27, 16]\) from the area between the absorber wall and metal shelves around \([30, 12]\).

**C. Baseline Methods**

We compare our tracking method against state-of-the-art with EMI and CNN-FP approaches:

**EMI.** We first detect NLOS [9] to exclude unreliable TOFs using a LOS quality indicator threshold (\(\beta \) of 1.5) obtained by analysis of the data. We combine the TOFs classified as LOS in a range-based PF. For a fair comparison, we implement a dynamic model to assure a fair comparison. A subsequent PF is not applicable in this case, as the trained CNN only produces a position estimate from the recorded data directly.

**CNN-FP.** We train a CNN-based position regressor [11], on the same data we use to train our GPR models. We use a 6 layer CNN architecture that consists of 4 convolutional layers for feature extraction and 4 fully connected layers for position estimation. The input tensor consists of the recorded complex CMs shifted by the estimated TOFs. We use ReLU activation for all layers, except for the last layer that uses a linear activation. We omit local pooling as this may harm the accuracy on time-series tasks [43]. Nevertheless, to enhance the receptive field of the CNN we enlarge the kernel sizes with the depth of the model: The first convolutional layer uses a kernel size of \([1 \times 3]\), the second \([2 \times 5]\), the third \([3 \times 15]\), and the last \([3 \times 30]\). As CNNs do not include dynamic information, we smooth the CNN position estimates using a linear Kalman filter (LKF) with a constant velocity dynamic model to assure a fair comparison. A subsequent PF is not applicable in this case, as the trained CNN only produces a position estimate from the recorded data directly.

First, we consider the magnitude kurtosis (KUR) (see Figs. 5a and 5b). For the full dataset (Figs. 5a), GPR learns a detailed spatial distribution for \(\mu_{j,f}\) with distinct areas of high (red/orange color) and low (green color) values. Outside of the recording area, \(\mu_{j,f}\) converges to a global mean of 0 due to the scaling. Based on this distribution different areas can be distinguished from each other, e.g., the area behind the absorber wall around \([x, y] = [27, 16]\) from the area between the absorber wall and metal shelves around \([30, 12]\).

**C. Baseline Methods**

We compare our tracking method against state-of-the-art with EMI and CNN-FP approaches:

**EMI.** We first detect NLOS [9] to exclude unreliable TOFs using a LOS quality indicator threshold (\(\beta \) of 1.5) obtained by analysis of the data. We combine the TOFs classified as LOS in a range-based PF. For a fair comparison, we implement a dynamic model to assure a fair comparison. A subsequent PF is not applicable in this case, as the trained CNN only produces a position estimate from the recorded data directly.

**CNN-FP.** We train a CNN-based position regressor [11], on the same data we use to train our GPR models. We use a 6 layer CNN architecture that consists of 4 convolutional layers for feature extraction and 4 fully connected layers for position estimation. The input tensor consists of the recorded complex CMs shifted by the estimated TOFs. We use ReLU activation for all layers, except for the last layer that uses a linear activation. We omit local pooling as this may harm the accuracy on time-series tasks [43]. Nevertheless, to enhance the receptive field of the CNN we enlarge the kernel sizes with the depth of the model: The first convolutional layer uses a kernel size of \([1 \times 3]\), the second \([2 \times 5]\), the third \([3 \times 15]\), and the last \([3 \times 30]\). As CNNs do not include dynamic information, we smooth the CNN position estimates using a linear Kalman filter (LKF) with a constant velocity dynamic model to assure a fair comparison. A subsequent PF is not applicable in this case, as the trained CNN only produces a position estimate from the recorded data directly.

First, we consider the magnitude kurtosis (KUR) (see Figs. 5a and 5b). For the full dataset (Figs. 5a), GPR learns a detailed spatial distribution for \(\mu_{j,f}\) with distinct areas of high (red/orange color) and low (green color) values. Outside of the recording area, \(\mu_{j,f}\) converges to a global mean of 0 due to the scaling. Based on this distribution different areas can be distinguished from each other, e.g., the area behind the absorber wall around \([x, y] = [27, 16]\) from the area between the absorber wall and metal shelves around \([30, 12]\).

**C. Baseline Methods**

We compare our tracking method against state-of-the-art with EMI and CNN-FP approaches:

**EMI.** We first detect NLOS [9] to exclude unreliable TOFs using a LOS quality indicator threshold (\(\beta \) of 1.5) obtained by analysis of the data. We combine the TOFs classified as LOS in a range-based PF. For a fair comparison, we implement a dynamic model to assure a fair comparison. A subsequent PF is not applicable in this case, as the trained CNN only produces a position estimate from the recorded data directly.
After deriving and selecting characteristic features that encode important spatial information, we now evaluate if they can be employed to enhance positioning results with our proposed framework in comparison with the baseline methods. For quantitative evaluation of the position accuracy we consider the statistics of the absolute position error (APE). Table 1 lists all the positioning results.

Fig. 6 shows the results of the EMI approach in the proposed scenario: Stable tracking with APEs in the low decimeter range (green color) is possible in most parts of the environment. However some areas are problematic, leading to a complete failure of the tracking filter, resulting in APEs in the higher meter range (yellow to red color). Interestingly, outside of these zones, the tracking filter converges again to a low APE. Therefore, due to low LOS anchor availability in problematic areas additional spatial information is required to maintain accurate tracking. Although a relatively low error median (MED) of 31.2 cm can be achieved (see Fig. 8, Table 1),

VIII. POSITIONING RESULTS

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.
TABLE I: Mean (MAE), Median (MED), 75th and 95th percentile (C75 / C95) of the APE in m.

|       | MAE    | MED    | C75   | C95   | ref. |
|-------|--------|--------|-------|-------|------|
| EMI   | 1.00   | 0.312  | 0.778 | 4.79  | Fig. 6 |
| CNN-FP-F   | 0.300  | 0.271  | 0.385 | 0.639 | Fig. 7a |
| CNN-FP-S   | 1.39   | 1.23   | 1.85  | 3.08  | Fig. 7b |
| GPR-F-F   | 0.409  | 0.276  | 0.467 | 1.35  | Fig. 7c |
| GPR-F-S   | 0.634  | 0.366  | 0.727 | 2.22  | Fig. 7d |
| GPR-AE-F   | 0.333  | 0.245  | 0.390 | 0.906 | Fig. 7e |
| GPR-AE-S   | 0.502  | 0.289  | 0.535 | 1.755 | Fig. 7f |

significant outliers occur in the areas with low LOS coverage that result in a high mean absolute error (MAE) of 1.00m and C95 of 4.79m. In relevant applications, such as robot or automatic industrial vehicle navigation or tool tracking, such deviations may lead to serious problems such as crashes with structures or the loss of equipment. However, these errors can be avoided by FP methods (i.e., the baseline CNN-FP and our proposed GPR method) as they exploit additional spatial information contained in the FP database.

For the full dataset CNN-FP can maintain accurate tracking in the complete environment (Fig. 7a). Due to full data availability, the CNN estimates all positions accurately, while potential outliers are smoothed out by the dynamics model of the LKF. This is also seen in the error statistics (Fig. 8, Tab. I): the MAE is 0.300m, errors below 38.5cm can be achieved in 75 % of the cases. However, a massive deterioration of the positioning accuracy occurs for the sparse dataset: CNN-FP lacks the necessary data in large parts of the environment and cannot properly extrapolate (Fig. 7b). As CNN-FP does not rely on observation modeling and only produces non-stochastic position estimates, the LKF tracking algorithm cannot adapt to this change in reliability.

Hence, in a typical application setting this reliability may remain undetected and may lead to system failure. This is also reflected in the error statistics (Fig. 8), where both the error median and 95th percentile (C95) increase significantly to 1.23m and 3.08m. Also, the MAE and C95 deteriorate to 1.39m and 3.08m. Note that in comparison with Fig. 6 CNN-FP suffers from these outliers in areas where enough LOS information is present to enable stable tracking for EMI. Our method compensates the shortcomings of both baselines by employing a statistical observation modeling that combines information sources, i.e., the ToF and FP database, in a PF.

Based on our feature selection, we compare the best-performing propagation feature GPs (GPR-P) and AE feature GPs (GPR-AE). Both GPR-P-F and GPR-AE-F significantly enhance the position accuracy over the EMI approach for the full dataset (Figs. 7b, 7c). In general, the GPR-AE-F performs better than GPR-P-F and yields a MAE of 0.330m and MED of 0.246m, which is lower than CNN-FP-F. However, GPR-AE-F returns outliers with higher errors (C95 of 0.906m) than CNN-FP-F. This may be due to a more problem-specific and detailed feature selection in the CNN, while our method compresses the information by feature extraction.

In contrast to the results on the full dataset, GPR-AE-F, unlike CNN-FP, show it full potential on the sparse dataset (Fig. 7d, 7e). In areas where no data recordings are available, the observation model components of the GP models produce high variance estimates, so that the estimated log-likelihood weights are mostly uniform. Thus, the majority of spatial information for the update of the PF is contributed by the LOS measurements, which, as seen in Fig. 6 are sufficient for accurate tracking. Hence, the performance decrease of GPR-AE-S is much lower than for CNN-FP (Fig. 8), so that for GPR-AE-S, the MED only decreases to 0.287m. However, in areas with low LOS anchor coverage significant outliers exist, most significantly in the area around 32, 15. Here, neither the sparse FP dataset nor the obstructed TOFs can provide valid spatial information. This causes an increased C95 of 1.60m. Thus, in general, with a spatially spare FP dataset, the position accuracy of GPR-AE-S is significantly higher than both baselines (CNN-FP-S and EMI).

In essence, our approach maintains stable tracking even in large environments with mixed propagation conditions. We only require a low amount of data recorded specifically in areas that are dominated by (diffuse) MPC. This drastically reduces the effort of data collection and maintenance.

IX. DISCUSSION AND LIMITATIONS

First, our feature extraction resorts to suitable propagation features known from the literature and leaves out other types of features, e.g., time-frequency-domain features. While this of course is far from being complete we leave a more thorough discussion of usable features to future work.

Second, we assume statistical independence between features in (5). Such independence might not necessarily be guaranteed by how we extract features using the autoencoder. Adapted versions of the autoencoder, i.e., disentangled variational autoencoders, that assure independent features may enforce this statistical independence. The statistical independence from the LOS component might be increased, e.g., by spectral subtraction.

Third, we might extract additional spatial information as the features we extract are only applied to the magnitudes of the CM – additional spatial information may be extracted from the phases. Also, we leave a study of how an additional pre-processing of the feature values might increase spatial consistency to future work. Moreover, the GPR in our framework builds upon stationary kernels, however, a domain-specific, non-stationary kernel might yield additional spatial information, like angular and radial dependence.
Fig. 7: Positioning errors of the FP approaches on the full (top row) and sparse (bottom row) datasets. The anchor positions are indicated with black triangles, while the black boxes indicate the positions of objects within the environment.

Fig. 8: Position error CDFs for 3 Anchors.

Fifth, we do not estimate weights for the information sources, so that all of them are treated equally. An additional optimal weighting most probably increases the positioning accuracy and makes the use of more features more viable. While we conducted a vast grid search to obtain a good representation, other methods such as evolutionary feature selection [46] or reinforcement learning [47] may yield more optimal feature combinations and/or weights.

X. CONCLUSION

We presented a novel two-stage tracking framework for channel measurement (CM) based positioning. It builds upon the extraction of characteristic features (both propagation- and autoencoder-based (AE)) and applies Gaussian process regression (GPR) on recorded fingerprinting (FP) data by training individual Gaussian processes (GPs) for subsets of our features. The GPs model observation likelihoods that represent spatial information. Hence, our approach allows for an information fusion using, e.g., a particle filter (PF) with state-of-the-art dynamics modeling and resampling. Most importantly, our approach works with spatially sparse datasets.

We evaluate our methods on data of a realistic industrial environment. We investigate both a spatially dense and smaller and sparse training data sets. Unlike CNN-FP our method adapts well to the sparse dataset and yields significantly more accurate positions. We attribute this performance gain to the fact that the framework models the reliability of the observation likelihood based on stationary, distance-dependent kernels, and therefore can also effectively integrate the spatial information provided by EMI approaches and rely on it in areas without available data to maintain accurate positioning.

ACKNOWLEDGEMENTS

This work was supported by the Bavarian Ministry for Economic Affairs, Infrastructure, Transport and Technology.
REFERENCES

[1] S. Aditya, A. F. Molisch, and H. M. Behairy, “A survey on the impact of multipath on wideband time-of-arrival based localization,” Proceedings of the IEEE, vol. 106, no. 7, pp. 1183–1203, 2018.
[2] D. Dardari, P. Closas, and P. Djuric, “Indoor tracking: Theory, methods, and technologies,” IEEE Trans. Veh. Tech., vol. 64, pp. 1263–1278, 2015.
[3] A. Molisch, “Ultra-wideband propagation channels,” Proceedings of the IEEE, vol. 97, pp. 355 – 371, 03 2009.

through the Center for Analytics-Data-Applications (ADA-Center) within the framework of “BAYERN DIGITAL II”. The authors also acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of “Souverän. Digital. Vernetzt.” Joint project 6G-RIC, project identification number: 16KISK020K.