Velocity-Based Channel Charting With Spatial Distribution Map Matching

Maximilian Stahlke, George Yammine, *Senior Member, IEEE*, Tobias Feigl, *Member, IEEE*, Bjoern M. Eskofier, *Senior Member, IEEE*, and Christopher Mutschler, *Member, IEEE*

***Abstract*—Radio fingerprinting (FP) technologies improve localization performance in challenging non-line-of-sight environments. However, FP is expensive as its life cycle management requires recording reference signals for initial training and when the environment changes. Instead, novel channel charting technologies are significantly cheaper. Because they implicitly assign relative coordinates to radio signals, they require few reference coordinates for localization. However, even channel charting still requires data acquisition and reference signals, and its localization is slightly less accurate than FP. In this article, we propose a novel channel charting framework that does not require references and dramatically reduces life-cycle management. With velocity information, e.g., pedestrian dead reckoning or odometry, we model relative charts. And with topological map information, e.g., building floor plans, we transform them into real coordinates. In a large-scale study, we acquired two realistic datasets using 5G and single-input and multiple-output distributed radio systems with noisy velocities and coarse map information. Our experiments show that we achieve the localization accuracy of FP but without reference information.**

***Index Terms*—Channel charting, fingerprinting (FP), machine learning, radio localization.**

I. Introduction

INDOOR localization serves as a key enabler for various downstream tasks in industrial production, health care, or networking [1]. Radio-based localization methods [2] are, beside other technologies such as camera [3], one of the most promising technologies for indoor localization [2]. If there are line-of-sight (LoS) conditions, systems based on time-of-arrival measurements [4] achieve subdecimeter localization accuracies. However, in realistic indoor environments, non-line-of-sight (NLoS) signals and multipath propagation degrade localization accuracy. In such environments, error mitigation methods such as NLoS identification [5] or error correction [6] enable robust localization only when redundant base stations (BSs) are available. Instead, fingerprinting (FP) also works in NLoS-dominated areas with few BSs [7], [8], [9]. However, FP is site-specific and its life-cycle management is expensive. Initial (supervised) training requires channel state information (CSI) measurements and corresponding reference positions. And as environmental changes alter fingerprints, updates are needed [8], [10]. Instead, channel charting on CSI implicitly models a manifold that represents the (local, relative) geometry of the environment [11]. The idea is that CSI are unique to a position in the environment and change smooth within it [12], [13]. And manifold learning projects these high-dimensional CSI onto 2-D coordinates that preserve the relative geometry. The most renowned approaches explicitly define the manifold using a distance matrix to model the radio geometry either through the CSI itself [12], [14], [15], [16], [17] or through physical motion models such as constant velocity [14]. Compared to FP, channel charting localizes less accurately [12] or requires unrealistic motion models [14]. Finally, for localization in real coordinates, it still requires few references and updates when the environment changes, so its life cycle management is expensive.

In this article, we address these weaknesses. Our key idea is that trajectory estimation on velocities is approximately error-free over short time horizons, allowing a Siamese neural network to derive a short-distance matrix and learn the global structure of the radio environment. We show how (noisy) velocity information, e.g., pedestrian dead reckoning (PDR) or odometry systems, improve the modeling of a channel chart. We also exploit topological map information to learn a transformation into real coordinates. This matching algorithm learns the spatial distribution of the map along with the transformation to adjust the orientation of the channel chart. So, for robust radio-based localization in the real world, we only need CSI, (noisy) velocity information, and a rough map representation of the area, e.g., a floor plan. We evaluate our framework on two radio systems, a 5G-based radio system and a distributed single-input and multiple-output (SIMO) system, and on erroneous velocities and map information. We show that our framework is independent of radio topologies and architectures and robust to noisy velocities and results in charts with the accuracy of FP but without expensive reference systems.

Manuscript received 4 March 2024; revised 17 May 2024 and 1 July 2024; accepted 3 July 2024. Date of publication 9 July 2024; date of current version 21 August 2024. This work was supported in part by the Fraunhofer Lighthouse project "6G SENTINEL" and in part by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt." joint project 6G-RIC (16KISK020K). *(Corresponding author: Maximilian Stahlke.)*

Maximilian Stahlke, George Yammine, Tobias Feigl, and Christopher Mutschler are with Fraunhofer IIS, Fraunhofer Institute for Integrated Circuits IIS, Division Positioning and Networks, 90411 Nuremberg, Germany (e-mail: maximilian.stahlke@iis.fraunhofer.de; george.yammine@iis.fraunhofer.de; tobias.feigl@iis.fraunhofer.de; christopher.mutschler@iis.fraunhofer.de).

Bjoern M. Eskofier is with the Department Artificial Intelligence in Biomedical Engineering (AIBE), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 91052 Erlangen, Germany (e-mail: bjoern.eskofier@fau.de).

Digital Object Identifier 10.1109/JISPIN.2024.3424768

The rest of this article is organized as follows. Section II discusses related work. Section III introduces our framework. Section IV describes our experimental setup. Section V presents numerical results. Section VI discusses limitations, and finally, Section VII concludes this article.

II. Related Work

This section discusses related work on applications, distance metrics, dimensionality reduction, and coordinate system transformation for channel charting.

Channel charting [11] learns the geometry of the radio environment and supports tasks such as user grouping [18], radio resource management [19], beamforming [20], or localization [12], [14], [21], [22], [23], [24], [25]. It typically runs in two phases. First, it estimates the distances between CSI that are proportional to the physical distance and models the manifold of the CSI. Second, it reduces high-dimensional CSI to a 2-D representation that reflects the coordinates of the radio environment.

Channel charting employs distance metrics that are based on the free-space path loss of radio signals [11], with extensions to make it insensitive to fast-fading effects [15]. Advanced approaches extract multipath components (MPCs) and exploit the path-loss for every MPC [26]. To utilize environmental information, Stahlke et al. [12] exploited MPCs from power-delay profile (PDP)s for time-synchronized high-bandwidth single-input and single-output radio systems. Stephan et al. [14] extended their metric for multiple-input and multiple-output (MIMO) radio systems to exploit phase information. However, distance estimates of radio signals are noisy due to limitations such as collinearity and bandwidth. Other approaches assume that channel measurements close in time are also close in space [23], [27], [28], [29]. However, this only holds for constant velocity models [14], [17].

Channel charting learns the 2-D manifold on high-dimensional channel measurements, where effective dimensionality reduction is crucial. There are nonparametric approaches such as principal component analysis [11], Laplacian eigenmaps [30], or Isomap [15]. However, they model nonlinearities inaccurately or not at all and do not generalize on unknown data. Therefore, channel charting typically employs parametric neural networks such as autoencoders [31], [32], Siamese networks [12], [14], [22], or triplet-based models [17], [23], [27], [28], [33] that generalize well and their nonlinearity reduces dimensionality effectively.

Channel charting only models the radio environment isometrically. For localization, a transformation from local geometry to global geometry and alignment to the coordinates of the real world is, therefore, necessary [22], [23], [24], [25]. This transformation is done with a few reference samples from the target coordinate system. And if the channel chart already represents a global geometry, a linear transformation is sufficient [12], [14]. However, to locate accurately, even channel charting requires updates as the environment changes [8]. Therefore, its life-cycle management effort is lower than with classic FP, but it is still not free. To completely avoid acquisition of reference measurements, Ghazvinian et al. [29] exploited additional map information. Their map information helps to align the channel chart coordinates with the environment. They define a map as a discrete probability density function (PDF) that represents the distribution of the recorded data to match the data distribution of the channel chart with the defined map through optimal transport. Ghazvinian et al. [29] inspire the map-matching component of our framework.

III. Method

This section introduces our novel channel charting framework. Section III-A introduces manifold learning of CSI. Section III-B describes how we exploit velocities to create a (relative, local) channel chart. Section III-C describes how coarse map information transforms the channel chart from local to global (real-world) coordinates. Section III-D describes our localization pipeline.

A. CSI on a Manifold

Radio systems measure the sum of all impinging signals from any direct link between transmitter and receiver and on multiple paths, e.g., from reflections in the environment. With the radio channel modeled as

$$h(t) = \sum_{n=0}^{N_\mathrm{p}-1} a_n \delta(t - T_n) \tag{1}$$

we can estimate the channel impulse response (CIR), i.e., CSI in time domain, by correlating a known bandwidth-limited measurement signal $s(t)$ with the (noise-free) received signal

$$\tilde{h}(\tau) = \int_{-\infty}^{\infty} y(t) s^*(t - \tau)\, \mathrm{d}t \tag{2}$$

where

$$y(t) = h(t) * s(t) \tag{3}$$

where N_p is the total number of signal paths, n is the index of the current path, a_n is the complex gain of the nth MPC, $\delta(\cdot)$ is the Dirac delta function, T_n is the delay of the component, and $*$ is the convolution operator. As the MPC delays depend on the position in space, the CIR can be defined as a smooth function of space. Thus, the CIR changes gradually in space and we can obtain a smooth transition function from the high dimensional manifold, defined by the CIR, onto the 2-D space.

B. Velocity-Assisted Channel Charting

The manifold is typically modeled by a distance matrix, i.e., a square matrix that contains all pairwise distances of CSI measurements. Modern techniques [8], [14] showed that we can obtain a globally valid distance matrix using CIRs and geodetic distances. However, they are inaccurate due to the nonconvexity of datasets and bandwidth limitations. Thus, we determine a sparse, but more accurate, distance matrix on additional velocity information to estimate distances between channel measurements. We can obtain velocity information from various sources. Especially in indoor environments such as automobile production lines or warehouse centers, robotic platforms or industrial vehicles often use wheel-based odometry [34], or

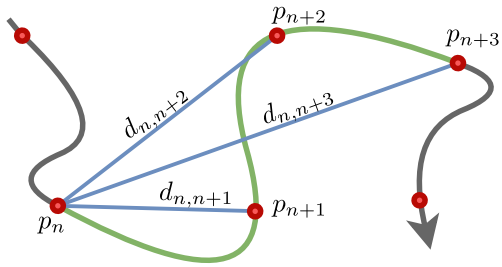


Fig. 1. Trajectory (gray line) with consecutive CSI measurements (red dots). The (blue) distances $d_{n,n+1}, \ldots, d_{n,n+3}$ are calculated within a window (green segment) for the positions $p_n, \ldots, p_{n+3}$.

pedestrians often use smartphones with inertial measurement units for velocity estimation [35]. Velocity information is prone to errors. Spinning wheels and fluctuating tire pressure worsen the odometry and incorrect step length estimation or drifting IMU signals worsen the velocity estimation of PDR algorithms. Therefore, integrating velocity to determine relative locations is error-prone and only practical for very short time. We exploit this fact to create a sparse low-distance matrix, which is then used by a Siamese network for channel charting.

1) Sparse Distance Matrix: As a tracking object moves along a trajectory in a given environment, its velocity and a radio system's CSI are periodically recorded. Velocity is integrated over a time window w, to estimate distances between the consecutive CSI. The distances of w are defined as

$$d_{n,k} = \|p_n - p_k\|_2 \tag{4}$$

$$= \left\| \int_{t=t_n}^{t_k} v(t)\,\mathrm{d}t \right\|_2 \tag{5}$$

where t_n and t_k are the time of the first and next measurement within w and $d_{n,k}$ is the Euclidean distance between the positions p_n and p_k estimated by the velocity $v(t)$. As the estimated trajectory drifts over time, we constrain the distance estimation so that $t_k - t_n < w$, while we calculate all distances from the first measurement at p_n to all consecutive measurements until the end of the window. Fig. 1 visualizes the distance estimation process. A tracking object, e.g., robot or pedestrian, equipped with an odometry or PDR, moves along a trajectory (gray line) within a radio environment and regularly records CSI (red). Within a certain time window (green segment) the (blue) distances between the position p_n and the consecutive positions $p_{n+1}, \ldots, p_{n+3}$, estimated by the velocity, are calculated. We slide w to the next position. The stride length s defines the number of measurements we skip. We use it to lower the amount of data for radio systems with high update rates.

2) Channel Charting: Both the CSI and the distance matrix are employed in channel charting to map the high-dimensional CSI of radio signals into a 2-D space that reflects the coordinates of the radio environment. Here, Siamese neural networks [36] are ideal to learn a transfer function from CSI to 2-D coordinates and generalize well on unknown CSI [12], [14], [22]. They lower the dimensionality of input data pairs and preserve a distance defined by a metric. And we use them as they also model the manifold of the underlying CSI data on a sparse-distance matrix. They minimizes the distance relation (4) by using the following loss function:

$$\mathcal{L}_\mathrm{d} = \beta \left| d_{n,k} - \|\boldsymbol{z}_n - \boldsymbol{z}_k\|_2 \right| \tag{6}$$

where $\boldsymbol{z}_n$ and $\boldsymbol{z}_k$ are the 2-D outputs of the neural network, i.e., the 2-D coordinates, given the CSI measurements of the radio system, $|\cdot|$ is the norm, and $\|\cdot\|_2$ the Euclidean distance. As the error increases with time within a window, distances to measurements farther from the start of the window are less reliable. Our weighting parameter β, therefore, changes linearly within the window in the range [1, 0) so that samples that are closer in time to the start of the window have a greater influence on the final result. Convolutional neural networks (CNNs) are known to efficiently extract the temporal and spatial correlations of the CSI and exploit them for robust localization, and therefore, dominate the state-of-the-art for (supervised) CSI FP [7], [12], [37], [38]. Thus, we design our Siamese network similarly [12]. It consists of four convolutional (conv.) layers for feature extraction, followed by two dense layers for discrimination. The kernel sizes are increasing [3, 5, 7, 10] with eight channels for every conv. layer. We apply batch normalization for the conv. layers followed by rectified linear unit activation functions to introduce nonlinearity. To reduce the dimensionality, we apply global average pooling after the last conv. layer. We employ the identity function as activation function for the last layer.

C. Adaptive Map Matching

Our velocity-assisted channel charts only represent the radio environment up to isometries. To exploit channel charting for localization, we must transform the local channel charts to the real-world coordinates. This transformation is typically performed with reference positions [12] or with map information of the environment [29]. The latter does not require any reference positions, but does require the exact spatial distribution of the channel chart. Instead, our adaptive approach learns the spatial distribution along with the transformation.

1) Optimal Transport: Ghazvinian et al. [29] derive a discrete PDF from map information, e.g., a floor plan, and match the channel chart coordinates to this distribution by optimal transport. They estimate a transportation matrix $\boldsymbol{T} \in \mathbb{R}_+^{s \times t}$, which satisfies the regularized optimal-transport function

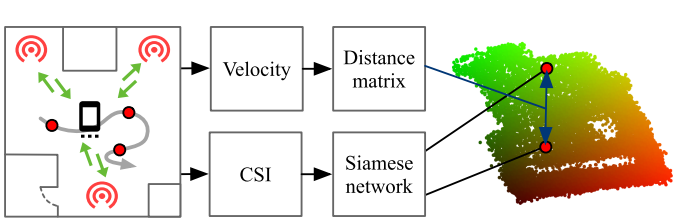
$$\Lambda(\boldsymbol{C}, \boldsymbol{p}, \boldsymbol{q}) = \underset{\boldsymbol{T} \in \gamma(\boldsymbol{p},\boldsymbol{q})}{\operatorname{argmin}} \langle \boldsymbol{T}, \boldsymbol{C} \rangle - \frac{1}{\lambda} H(\boldsymbol{T}) \tag{7}$$

where $\langle \cdot, \cdot \rangle$ is the Frobenius inner product, $\boldsymbol{p}$ and $\boldsymbol{q}$ are probability distributions of samples from the source domain Ω_s (in our case the distribution of the channel chart) and the target domain Ω_t (the distribution of the topological map). Their joint probability is γ and $\boldsymbol{C} \in \mathbb{R}_+^{s \times t}$ is a distance matrix, calculated using the Euclidean distance, between the channel chart coordinates and the map samples. It describes the cost to transport probability mass between the channel chart and map domain. The solution for (7) is

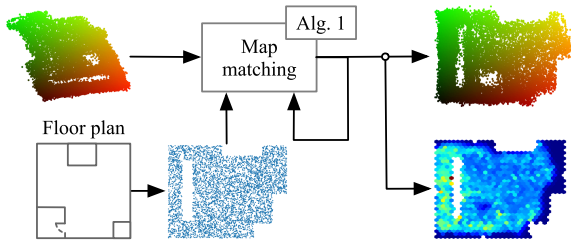
$$\boldsymbol{T} = \operatorname{diag}(\boldsymbol{a}) \boldsymbol{K} \operatorname{diag}(\boldsymbol{b}) \tag{8}$$

where the Sinkhorn–Knopp algorithm [39] calculates $\boldsymbol{K} = [\mathrm{e}^{-\lambda C_{i,j}}], i = 1, \ldots, s, j = 1, \ldots, t, \boldsymbol{a} \in \mathbb{R}_+^s$ and $\boldsymbol{b} \in \mathbb{R}_+^t$

$$\boldsymbol{a} \leftarrow \boldsymbol{p} \oslash \boldsymbol{K}\boldsymbol{b} \quad \text{and} \quad \boldsymbol{b} \leftarrow \boldsymbol{q} \oslash \boldsymbol{K}^\top \boldsymbol{a} \tag{9}$$



(a) Stage 1: Channel charting.



(b) Stage 2: Map matching.

Fig. 2. Two stages of our localization pipeline. Stage 1 (left) generates a channel chart based on velocities and CSI. Such a chart only represents the radio geometry up to isometries. Stage 2 (right) learns a linear transformation to the real world coordinates on map information for localization.

where $\cdot^\top$ is the transpose of the matrix and $\oslash$ is the element-wise division operation. The algorithm iteratively estimates (7), while λ regularizes the stability of the convergence by controlling the entropy $H(\boldsymbol{T})$. The higher the entropy $H(\boldsymbol{T})$, the faster the convergence but also the less optimal the transport between the probability distributions $\boldsymbol{p}$ and $\boldsymbol{q}$ is. As the algorithm only consists of linear operations, it can be differentiated and we can formulate it as loss function

$$\mathcal{L}_\mathsf{m} = \langle \boldsymbol{T}, \boldsymbol{C} \rangle \tag{10}$$

to minimize the difference of the distribution of the channel chart coordinates and the distribution of the topological map.

2) Adaptive Map Distribution: Ghazvinian et al. [29] assume that they know the distribution of the source domain $\boldsymbol{p}$ or assume a uniform distribution within the map. These assumptions only apply to situations in which a locatable object moves in straight lines at a constant speed. However, these (uniform distribution) assumptions are not realistic in indoor environments with (un)intentionally (temporarily) inaccessible areas where no CSI measurement exists and their map matching fails. In contrast to Ghazvinian et al. [29], we employ learnable probabilities within the discrete map distribution.

3) Map Matching: Inspired by Ghazvinian et al. [29], who learn manifold and map matching simultaneously, we formulate a two-stage optimization approach to decouple the channel charting from the map matching. This lowers the complexity in the optimization process, and the nonideal map does not influence the distribution of the channel chart.

In the first stage, we estimate the channel chart that only represents the geometry of the environment. In the second stage, we learn a linear transformation to match the channel chart to the map. Algorithm 1 shows the pseudocode of our map-matching algorithm. We feed the channel chart coordinates of the training data X_s of size N_s, the samples of the topological map Y_t of size N_t, and the initial rotation ϕ_i to the algorithm. We select 2-D coordinates within the map area, e.g., from a floor plan, to obtain the map samples Y_t; see Section IV-E. We assign a probability to each coordinate. q_ζ represents the PDF of the map coordinates. The probabilities are initialized uniformly for all samples. Similar to the channel chart coordinates X_s, each sample is assigned a uniform probability, resulting in the channel chart PDF p. Algorithm 1 (line 1) shows the translation between the channel chart X_s and the map Y_t, which is initialized to match their centers of mass. Then (line 3), we run our optimization for

Algorithm 1: Map Matching.

Input: $X_\mathsf{s} = \{x_i\}^{N_\mathsf{s}} \in \Omega_\mathsf{s}, Y_\mathsf{t} = \{y_i\}^{N_\mathsf{t}} \in \Omega_\mathsf{t}, \phi_i$
Output: Linear transformation, i.e., t_θ, ϕ_κ
Init: $q_\zeta = \mathcal{U}^{N_\mathsf{t}}, \phi_\kappa = \phi_i$

1 $t_\theta = \text{InitTranslation}(X_\mathsf{s}, Y_\mathsf{t})$;
2 **for** I_iter *iterations* **do**
3 $B = \text{batches}(X_\mathsf{s})$;
4 **foreach** b *in* B **do**
5 $b = \text{linTrans}(b, t_\theta, \phi_\kappa)$;
6 $\mathcal{L}_\mathsf{m} = \text{Sinkhorn}(b, Y_\mathsf{t}, q_\zeta)$
7 $t_\theta \leftarrow t_\theta + \eta \frac{\partial \mathcal{L}_\mathsf{m}}{\partial t_\theta}$;
8 **if** $k > I_\mathsf{wt}$ **then**
9 $\phi_\kappa \leftarrow \phi_\kappa + \eta \frac{\partial \mathcal{L}_\mathsf{m}}{\partial \phi_\kappa}$;
10 **end**
11 **if** $k > I_\mathsf{wl}$ **then**
12 $q_\zeta \leftarrow q_\zeta + \eta \frac{\partial \mathcal{L}_\mathsf{m}}{\partial q_\zeta}$;
13 **end**
14 **end**
 end

I_iter epochs on the channel chart coordinates, which are split into B batches due to memory limitations. The translation t_θ and the rotation ϕ_κ are applied to the coordinates of the channel chart (line 5) and the Sinkhorn distance is estimated (line 6). Next (line 7), we first minimize the Sinkhorn distance w.r.t. the translation t_θ for I_wt periods to ensure the translation t_θ converges before we start to optimize the rotation ϕ_κ. Once, the parameters of the linear transformation are estimated using the static map, i.e., $I_\mathsf{wl}, I_\mathsf{wl} > I_\mathsf{wt}$ periods, the probabilities of the map distribution q_ζ are also optimized to adapt the map to the data distribution. Our (trained) channel chart represents the actual data distribution of the real environment. Instead, the coarse map only represents a high-level perspective of this environment. Thus, our goal is for the map to adapt to the channel chart through our map matching algorithm. Therefore, there is no benefit in learning the probabilities of the channel chart p. Our experiments showed that convergence is very sensitive to the initial rotations ϕ_i and often ends in local minima or the channel chart confuses the x- and y-axes. To circumvent these weaknesses, we repeat our map matching algorithm several times to select the (error-free) transformation parameters with the smallest Sinkhorn distance.

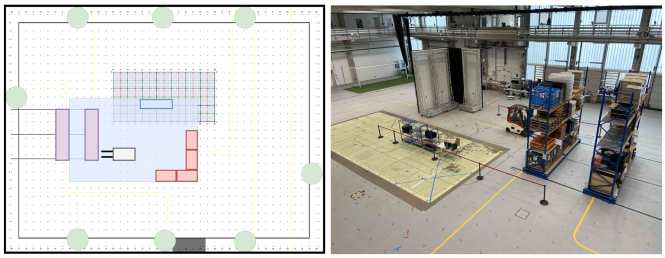


Fig. 3. Schematic top view (left) of the environment (right) of our 5G dataset with reflective walls (red rectangles), BSs (green dots), small shelves (blue rectangle), large shelves (purple rectangles), and the (blue) recording area (20 m × 10 m).

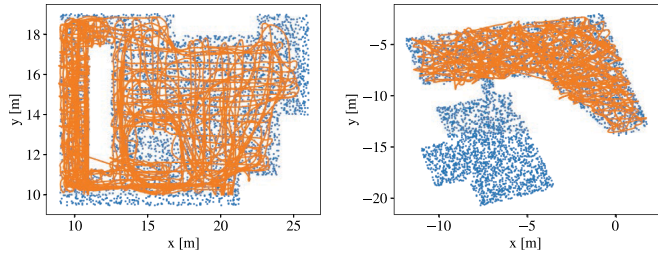


Fig. 4. Topological maps of the 5G (left) and the SIMO (right) datasets (blue: discrete coordinates; orange: real training trajectories).

D. Localization Pipeline

Fig. 2 shows our localization pipeline, which we split into two phases: 1) the channel charting phase [see Fig. 2(a)] and 2) the transformation phase [see Fig. 2(b)].

1) In the channel charting phase, one or more locatable objects move within the same environment. They are equipped with an odometry or PDR system to determine their velocities. The locatable objects also communicate with a radio system (red) to determine location-specific CSI (red dots). In a postprocessing step, we generate a sparse distance matrix between the CSI measurements from the velocity information; see Section III-B2. By learning the distances for given measurements, the parametric Siamese network learns the manifold of radio signals, i.e., the relative 2-D coordinates of CSI.
2) In the transformation phase [see Fig. 2(b)], we transform these relative 2-D coordinates into the real coordinates to enable localization with channel charts. Our map matching algorithm requires a floor plan of the area as a discrete PDF (blue dots). The map also covers areas where measurements are impossible, e.g., outside the range of the radio system or blockage due to objects. Our map-matching algorithm learns a linear transformation that fits both distributions and at the same time learns the probabilities of the map [heatmap in the bottom right of Fig. 2(b)] to fit the data distribution of the training data.

IV. Experimental Setup

This section presents our 5G and SIMO data collection setup, preprocessing of the datasets, and generation of corresponding velocities and topological maps. The 5G and SIMO datasets differ in the type of radio signals and the type of movement.

A. 5G Radio System

To record our 5G dataset, we use an experimental 5G uplink time-difference-of-arrival (TDoA) setup with eight commercial off-the-shelf software-defined-radio BSs. The radio system has a center frequency of 3.75 GHz with a bandwidth of 100 MHz. The BSs are synchronized by a signal generator. We record the time-domain CIR data at a frequency of 100 Hz. In parallel, we record synchronized reference positions with millimeter accuracy using an optical Nikon iGPS reference system. Fig. 3 (left) shows the schematic top view of the environment of the real-world setup (right). The BSs (green dots) are placed at the edges of the environment. The environment emulates a small industrial setup with large shelves (purple rectangles), a working desk (blue rectangle), large reflective walls (red rectangles), and a forklift (gray rectangle). The reflective walls block the signals that impinge on their backside. Near the walls, most BSs are in NLoS to the transmitter. We use a commercial mobile phone with a directional antenna as a transmitter. In the study, a person carries the phone in their hand directly in front of them to imitate a pedestrian using the mobile phone while walking. Note that this person may shadow the 5G signals w.r.t. their view point. The training dataset consists of 150 849 samples; see Fig. 4 (left, orange). The natural movement of the person with different standstill moments leads to a mean velocities of 0.94 m/s and a standard deviation (std. dev.) of 0.46 m/s. We record 18 256 test data on separate, independent trajectories, e.g., from another pedestrian; see Fig. 8 top left.

B. SIMO Dataset

To study the impact of distributed SIMO radio systems [40] on channel charting performance, we use a public dataset [41]. The orthogonal frequency-domain modulation radio system consists of four BSs with 2 × 4 antenna arrays each. The center frequency is 1.272 GHz with a bandwidth of 50 MHz. All antennas are over-the-air synchronized in frequency, time, and phase. The transmitter is equipped with an omnidirectional antenna. We study the industrial dataset [41] that places the antenna arrays in pairs close to each other at the edges of an L-shaped recording area in the surroundings of a research factory campus. The recording area has a size of 11 m × 13 m, with a metal container that causes NLoS and multipath propagation. Data recording was done by a robotic platform and the reference positions were estimated by an accurate tachymeter. The training (see Fig. 4 right, orange) and test (see Fig. 9 top left) datasets have 59 137 and 23 478 samples. The CSI data are in the frequency domain (frequency responses). The robot moves slower and more smoothly than pedestrians, 0.28 m/s in the training data and 0.25 m/s in the test data on average, with a standard deviation of only 0.10 m/s.

C. Preprocessing

As input to our framework, we exploit the CSI in the time domain, i.e., CIRs. For a measurement snapshot i at BS k, the CIR of length T is given in vector form as

$$\tilde{\boldsymbol{h}}_i^{(k)} \stackrel{\text{def}}{=} \left[\tilde{h}_i^{(k)}(0), \ldots, \tilde{h}_i^{(k)}(T-1)\right] . \tag{11}$$

TABLE I
PARAMETERS OF OUR NOISE LEVELS: BIAS IN THE ANGULAR VELOCITY (ANG. BIAS), BIAS IN THE MAGNITUDE OF THE VELOCITY (MAG. BIAS), AND HEADING JUMPS (HEAD. JUMP)

		ang. bias rad/s	mag. bias m/s	head. jump (pos. / ang.)
Noise level	0	-	-	-
	1	-	-	$[5, \frac{\pi}{4}]$, $[20, -\frac{\pi}{8}]$, $[60, -\frac{\pi}{6}]$
	2	$\frac{\pi}{200}$	-	$[5, \frac{\pi}{4}]$, $[40, -\frac{\pi}{8}]$, $[60, -\frac{\pi}{6}]$
	3	$\frac{\pi}{100}$	-	$[5, \frac{\pi}{4}]$, $[40, -\frac{\pi}{8}]$, $[60, -\frac{\pi}{6}]$
	4	$\frac{\pi}{200}$	-0.1	$[5, \frac{\pi}{4}]$, $[40, -\frac{\pi}{8}]$, $[60, -\frac{\pi}{6}]$

Head. Jump is defined by a position within the trajectory, as percentage of the trajectory and an angle [Pos., Ang.].

As the SIMO dataset contains CSI in the frequency domain, we apply the inverse Fourier transform to obtain time-domain equivalent CIRs

$$\tilde{\boldsymbol{h}}_i^{(k)} \overset{\text{def}}{=} \mathscr{F}^{-1}\left\{\tilde{\boldsymbol{\hbar}}_i^{(k)}\right\} \tag{12}$$

where $\mathscr{F}\{\cdot\}$ is the Fourier transform operator and $\tilde{\boldsymbol{\hbar}}_i^{(k)}$ the CSI in the frequency domain. Given the CSI in the time domain, we use the preprocessing scheme of Stahlke et al. [12]. Since our approach does not rely on phase synchronization, employ the PDP of the CIR. We generate a 2-D input tensor of dimensions $N_{\text{BS}} \times L_{\text{w}}$, with the number N_{BS} of BSs and L_{w} samples of the PDPs. For the 5G system, we pad the PDPs by the TDoA of the first arriving path to compensate the relative time alignment of the impinging signals. As the PDPs data of the SIMO system are already synchronized in time, we do not pad them.

D. Velocity Simulation

Velocity estimation based on signals from odometry or PDR systems is prone to errors due to spinning wheels, varying tire pressure, incorrect step length estimation, and drifting IMU sensors. For practical reasons, we focus on PDR error sources. Inspired by the findings of Jimenez et al. [42], we focus on the most serious sources of error, heading drift, absolute heading error, and incorrect step length estimation. We simulate three different sources of error using the (derived) reference velocities from recorded reference positions. Table I summarizes the sources of error as follows:

1) bias of the angular velocity (ang. bias) of the heading, e.g., due to a drifting gyroscope;
2) bias of the magnitude of the velocity (mag. bias), e.g., due to incorrect step length estimation;
3) heading jumps (head. jump), e.g., by exceeding the sensitivity limits of a gyroscope.

We divide the sources of error into different noise levels with increasing levels of error. Fig. 5 shows exemplary effects of noise levels. For clarity, we demonstrate the influence of noise levels on trajectory estimation on the 5G dataset. Fig. 5 shows the complete trajectory (blue), a reference segment of 60 s (orange), and estimated trajectory based on the noisy velocities (green). Level (0) is noise-free, i.e., the velocity estimates are error-free (orange segment). Level (1) contains different jumps in the heading and does not distort the angular velocity. Level (2) again contains different jumps in the heading and slightly distorts the angular velocity. Level (3) doubles the angular velocity deviation. Level (4) contains various heading jumps, slightly distorts the angular velocity and distorts the magnitude.

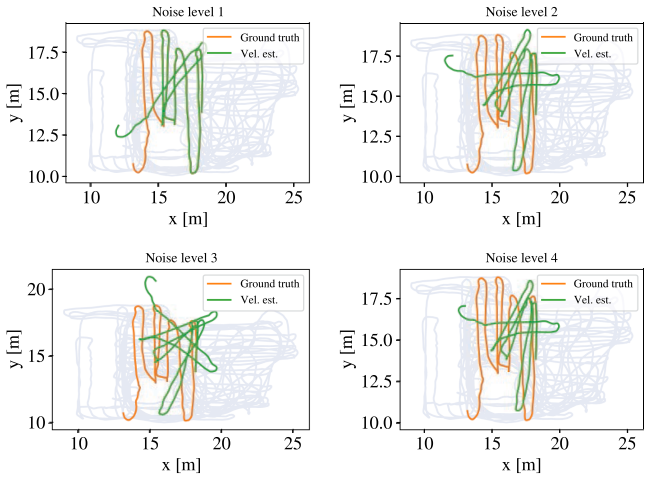


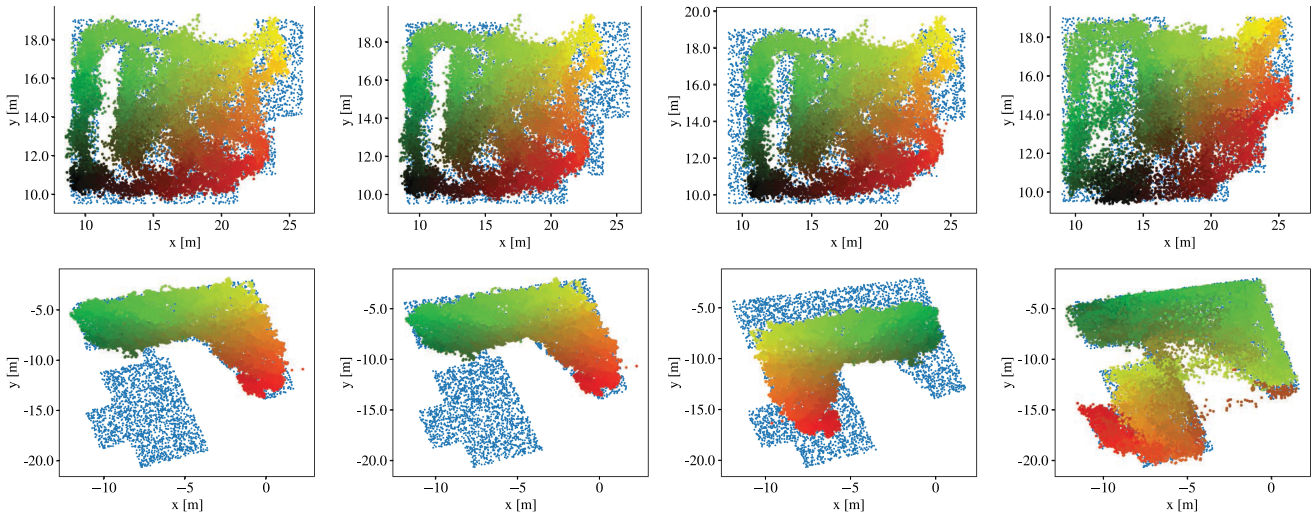
Fig. 5. Examples of velocity-assisted trajectories at four different noise levels (from lowest to highest). (Bue: Full reference trajectory; orange: 60 s segment of it; and Green: Estimated trajectory based on noisy velocities).

E. Maps

To learn a transformation from the local channel chart to the real world-coordinates, we exploit map information; see Section III-C. The map information consists of discrete locations within a topological map, and there must be a clear match of the shape of the channel chart with the shape of the area of the map; compare Fig. 4, which shows the 5G (left) and SIMO (right) datasets, the map coordinates (blue), and trajectories (orange). We found that 5000 samples provide accurate localization in both environments. For each position, there is a probability that we may map channel chart coordinates there. We assign a uniform probability to all map coordinates as the spatial density of our channel charts is unknown. The probability that a trackable object is located at a certain coordinate is the same for all map coordinates. The 5G map is limited to the recording area (blue); see Fig. 3. Our mapping considers all static objects in the area and excludes the dynamic forklift. We derive the SIMO map from reference coordinates. For a fair comparison, we simulated the additional unknown area (bottom left) with no (orange) reference trajectories.

V. EVALUATION

In the following, we evaluate our framework (see Section V-B), followed by an ablation study on the effects of varying window sizes and velocity noise levels (see Section V-C) and the influence of the numbers of BSs (see Section V-D). We train all methods for 10 000 epochs to ensure their convergence. Step lengths of $s = 10$ for the 5G dataset and $s = 2$ for the SIMO dataset enable fair comparison. The training data generates and optimizes the FP baseline and variants of channel charting frameworks. We evaluate all methods on an unknown separate



(a) Least sq. (b) Ours (c) Static map (d) Combined

Fig. 6. Results of various map matching algorithms on the 5G (top row) and SIMO (bottom row) datasets, from left to right: The least-squares approach, *Our* approach, map matching with a static map, and the combined approach. Blue points show the map distribution and the gradient points show the channel chart, w.r.t. the reference. Compare upper left corner in Figs. 8 and 9 for the ground-truth trajectories.

test dataset. As an error metric, we use the 90th percentile of circular Error (CE90), i.e., the Euclidean distance from the estimated position to the reference position.

A. Baselines

For a fair evaluation, we compare our novel channel charting framework (Ours) to various baseline methods. We directly compare our velocity-assisted channel charting to an FP baseline. Based on Stahlke et al. [8], FP employs the same model as the Siamese network and optimizes the output directly on the reference coordinates of the CSI values using the Euclidean-distance loss. Inspired by Stahlke et al. [12], the (Least sq.) baseline transforms our velocity-assisted channel chart to real-world coordinates using least-squares optimization on the references of the training data. Two additional baselines leverage our velocity-assisted channel charting, but use different map matching algorithms. The baseline (static map) is similar to our map matching algorithm but learns the map's probabilities along with the linear transformation. Instead, Ghazvinian et al.s' [29] baseline (Comb.) learns the manifold simultaneously with the map matching algorithm. For all baselines we employ the same Siamese network, see Section III-B2. To optimize it, we apply the combined loss $\mathcal{L}_{\mathsf{comb}} = \mathcal{L}_{\mathsf{m}} + \mathcal{L}_{\mathsf{d}}$.

B. Velocity-Assisted Channel Charting With Map Matching

This evaluation examines velocity-assisted channel charting under realistic conditions (velocity noise level 3) together with our map matching algorithm using topological map information. To do this, we use a window size of 15 s for the 5G dataset and a window size of 30 s for the SIMO dataset. We use $I_{\mathsf{iter}} = 150$ iterations for training, with $I_{\mathsf{wt}} = 50$ and $I_{\mathsf{wl}} = 100$ for the warm-up periods and $\lambda = 30$. We found that a batch size of 3000 samples is large enough to represent the data distribution.

TABLE II
LOCALIZATION ERRORS (CE90) ON THE 5G AND SIMO DATASET OF OUR ADAPTIVE APPROACH (OURS), STATIC MAP MATCHING (STATIC MAP), THE LEAST-SQUARES LINEAR TRANSFORMATION (LEAST SQ.), AND THE COMBINED APPROACH (COMB.)

Type	FP	Least sq.	**Ours**	Static map	Comb.
5G	0.90	0.92	1.16	1.71	3.43
SIMO	0.56	0.71	0.90	9.65	11.08

The map matching algorithm is repeated for 60 times, with 20 different equidistant starting rotations in the range $[0.2\pi)$ and inversions of the x and y axes. We selected the transformation with smallest Sinkhorn distance for the final result. Table II lists the results. Our approach (Ours) yields similar results as the FP baseline: CE90 of 1.16 m for the 5G and 0.90 m for the SIMO dataset. The error increases primarily due to the linear transformation to the coordinates of the real world. Similar to FP, the least-squares transformation achieves a CE90 of 0.92 m for the 5G and 0.71 m for the SIMO dataset. Fig. 6, top row from left to right, shows the map matching of the 5G dataset for the least-squares approach, our approach, the static map approach, and the combined approach. The gradient shows the channel chart coordinates w.r.t. the position of the reference coordinates; see Fig. 8 in the upper left corner. Compared to FP (CE90 = 0.26 m), our approach yields a higher CE90 of 1.16 m due to rotation errors of the map matching. The static map yields a rightward shift error, a CE90 of 1.71 m, as the (simplified) assumption that the data is uniformly distributed within the map is incorrect. The combined approach suffers from similar problems and results in an even higher CE90 of 3.43 m. While the coordinates of the channel chart closely match the area of the map, the Siamese network matches the distribution of the map. Therefore, points on the left side are moved to the right to match the map distribution. Instead, our approach learns the distribution of probability mass within the

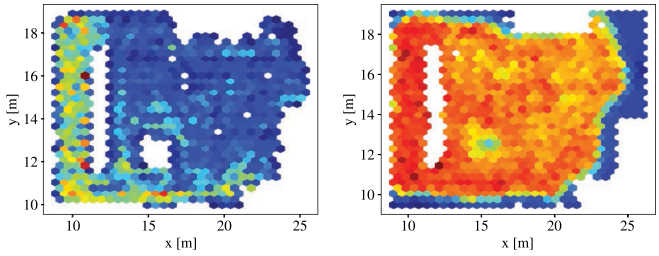


Fig. 7. Distribution of the 5G dataset within the environment (left) and the learned low (blue) and high (red) map probabilities (right).

TABLE III
LOCALIZATION RESULTS (CE90) OF CHANNEL CHARTING W.R.T. NOISE LEVELS AND WINDOW SIZES ON THE 5G AND SIMO DATASETS

Radio system	Window size [s]	Noise level / Error (CE90 in m)				
		0	1	2	3	4
5G	5	1.41	6.34	6.58	6.55	1.79
	15	0.85	0.89	0.90	0.92	0.94
	30	0.88	0.92	0.88	0.98	0.95
	60	0.87	0.87	0.96	1.11	0.98
SIMO	5	4.73	3.96	4.98	4.76	4.66
	15	3.39	0.69	3.56	2.79	2.79
	30	0.54	0.56	0.61	0.71	0.71
	60	0.54	0.56	0.71	0.94	0.91

map; see Fig. 7 on the right (blue: low probability; and red: high probability). Fig. 7 on the left shows the data distribution of the channel chart. We can clearly see that the probability mass fits well to the distribution of the data. All areas outside the training area are assigned a low probability in the map. The area on the left has a higher probability as in the data distribution. The bottom row in Fig. 6 shows similar behavior in the SIMO evaluation. Our method yields smaller rotation errors compared to the least-squares approach, while the static map method fails completely. The combined approach achieves poor localization performance (CE90 = 11.08 m) as it tries to uniformly adjust the channel chart coordinates within the map. Our approach mitigates this by learning that the area does not contain any data points.

C. Effects of Noise Level and Window Size

This ablation study investigates the effects of erroneous velocity estimates and different window sizes on the performance of velocity-assisted channel charting. To do so, this time, we evaluate channel charting isolated from map matching and use the references of the training data for the linear transformation into the real coordinates. Our framework is evaluated on our five different noise levels and four different window sizes for 5, 15, 30, and 60 s. For training, we use eight BSs of the 5G setup and three BSs of the SIMO setup. The results (see Table III) show that the window size significantly influences the modeling of the manifold of the CSI data. Velocity-assisted channel charting fails for both datasets with a window size of 5 s. The 5G dataset achieves higher accuracies with window sizes about 15 s. The SIMO dataset requires at least a window of 30 s to achieve an adequate localization. This indicates that window size correlates with movement speed. As the average

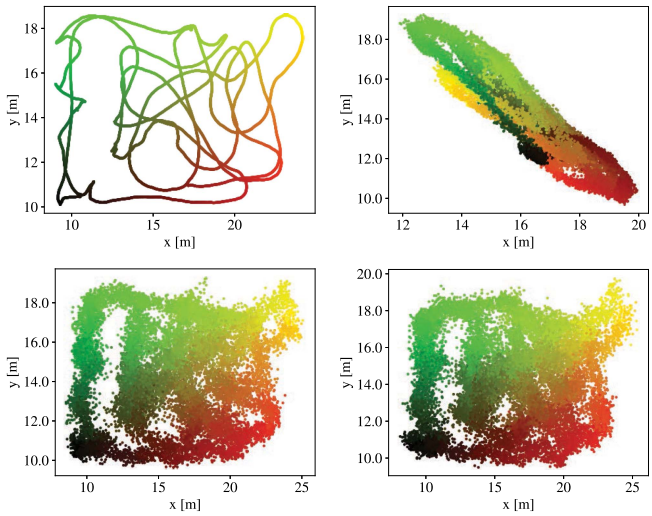


Fig. 8. Channel charts generated on different time windows of the 5G dataset (top left corner: Reference trajectory of the test dataset; top right: channel chart for 5 s; bottom left: 15 s; bottom right: 60 s).

velocity of the SIMO dataset (0.28 m/s) is significantly lower than that of the 5G dataset (0.95 m/s), the estimated distances of the SIMO dataset are significantly smaller. Thus, the Siamese network cannot model a globally valid channel chart based on local distances. Fig. 8 shows the reference of the test trajectory for the 5G test dataset (top left corner) and the channel charts for 5 s (top right), 15 s (bottom left), and 60 s (bottom right) window sizes. All channel charts reveal good spatial consistency, except 5 s that do not represent the global geometry of the test data. Channel charting achieves the best results with a window size of 15 s, (CE90 of 0.85 m to 0.94 m for the noise levels) on par with the FP results (CE90 = 0.90 m). The results deteriorate with a window size of 60 s as the noise level increases from CE90 = 0.87 m (noise level 1) to 1.11 m (noise level 3). We think that this is caused by the accumulation of errors of the noisy velocity information. The larger the window, the greater the errors in the distance estimates and the worse the performance of channel charting. However, the window size must be sufficiently large to restore the global structure of the environment. We see similar results in the SIMO evaluation in Fig. 9. The reference trajectory is shown top left in Fig. 9 is well represented by the channel chart for a window size of 30 s (bottom left), while the chart fails on a 5 s window (top right). At noise level 1 and a window size of 60 s, a CE90 of 0.56 m corresponds to the accuracy of FP (0.56 m). Accuracy decreases with higher noise levels up to 0.94 m.

D. Effects of Varying Numbers of BSs

This ablation study assesses the effects of different numbers of BSs on the performance of channel charting. We use a 15 s time window for the 5G and a 30 s time window for the SIMO dataset with noise level 0. Again, we use the references of the training data for the linear transformation into the real coordinates. Table IV lists the results. We use the TDoA values from at least two BSs of the 5G radio system to create the input tensors for the Siamese network. The SIMO setup consists of up to four BSs. The results show that our velocity-assisted channel

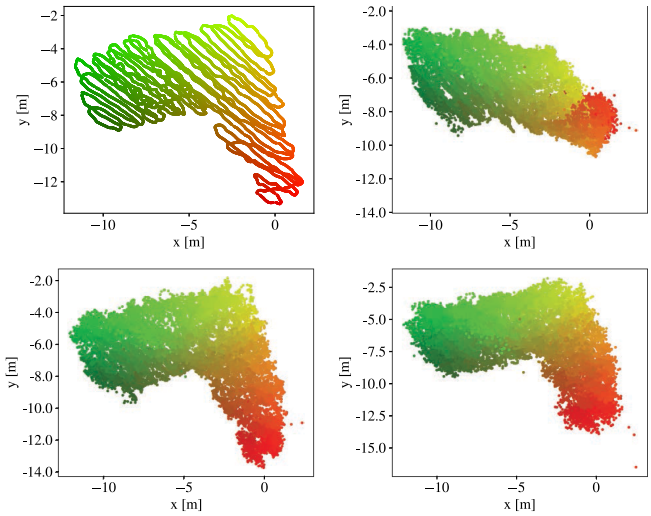


Fig. 9. Channel charts generated on different time windows of the SIMO dataset (top left corner: Reference trajectory of the test dataset; top right: Channel chart for 15 s; bottom left: 30 s; bottom right: 60 s).

TABLE IV
LOCALIZATION RESULTS FOR THE CHANNEL CHARTING (CC) COMPARED TO FP FOR DIFFERENT NUMBERS OF BSS

		# BSs / Error (CE90 in m)							
		1	2	3	4	5	6	7	8
5G	CC	-	4.95	1.30	1.17	1.26	0.97	0.96	0.85
	FP	-	2.67	1.28	1.12	1.38	1.05	0.99	0.90
SIMO	CC	1.24	0.64	0.54	3.25	-	-	-	-
	FP	1.13	0.68	0.56	0.52	-	-	-	-

charting achieves similar results to FP. In the 5G setup with two BSs, the localization accuracy of FP is higher than that of channel charting, as the Siamese network may not have enough information to model the manifold. However, for more BSs, channel charting works similarly to FP. On the SIMO dataset, the performance of channel charting is similar to fingerprint except for four BSs. Fig. 10 shows the reference locations (left) and the channel chart (right). The chart shows good spatial consistency as the gradient is well recovered. However, in the region where $x < -5$, the channel chart appears to be skewed about the x-axis. We determine the average of the Pearson correlation coefficients (PCCs) of the CSI for all BS combinations including all antennas of each BS. We found that there are strong correlations between the neighboring BS pairs, with a PCC of 0.86 and 0.73. While the BSs are very similar in the signal space, they are placed in different locations in the environment. Due to this redundancy, we suspect that the manifold is ambiguous and has multiple representations in 2-D space. When comparing the FP results for three and four BSs, we also cannot see any significant improvement. By removing a BS, our Siamese network converges reliably as it breaks the symmetry and redundancy of the data.

VI. LIMITATIONS

Our experiments show that our velocity-assisted channel charting algorithm achieves similar accuracies to FP, although it is independent of radio topologies and architectures and does not require expensive reference positions. However, we believe that the independent velocity estimation trajectories must cover a significant portion of the environment to achieve global consistency of the channel chart. This also means that the time windows have to be longer at low velocities. Consequently, for the 5G dataset, time windows of 15 s are sufficient due to the high velocities, while for the SIMO dataset, at least 30 s are required due to low velocities.

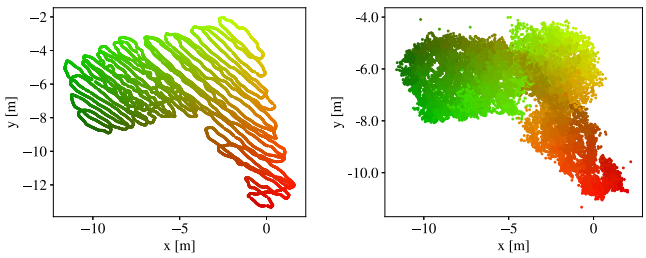


Fig. 10. Channel charts with four BSs of the evaluated MIMO radio systems. The estimated channel chart is displayed on the right. The color gradient shows the spatial consistency of the positions.

To transfer the local channel charting coordinates to the real environment, we use map information that reflects the rough geometry of the environment. Although we achieve higher localization accuracy compared to the state-of-the-art, we require a clear match of the shape of the channel chart within the map. This limits map matching to rotation-invariant and translational map information and, does not work for, e.g., (rotation-variant) rectangular or circular areas without unique features such as shelves or worktables. However, as we only use channel charting in NLoS-dominated areas, we can assume enough structure due to signal blocking by obstacles.

Our map matching algorithm is limited to linear transformation with rotation and translation. However, we cannot learn scaling errors due to incorrect step estimation (noise level 4) along with the map's probabilities. The map matching algorithm shrinks the channel chart and assigns it to an arbitrary area while it learns that the surrounding areas have no data assigned to them. Therefore, the scale of the channel chart must be estimated before map matching.

VII. CONCLUSION

To compensate for the overhead of managing the life cycle of state-of-the-art FP and channel charting technologies in realistic applications, we propose a framework that enables quasi-unsupervised indoor localization using (5G) radio systems without reference positions. Our velocity-assisted channel charting approach with adaptive map matching achieves accuracies of up to CE90 = 1.16 m for a 5G and 0.90 m for an SIMO radio system, similar to supervised FP, even with noisy velocities and coarse map information. Therefore, our approach is applicable to low-cost sensor systems such as smartphone-based PDR or odometry of robotic platforms in combination with CSI recordings. Our adaptive map matching employs topological map information such as floor plans to learn a transformation of the local channel chart coordinates to the real environment. In contrast to the state-of-the-art, our map matching algorithm does not require reference coordinates but only a coarse representation of the

environment, as it learns to adjust the map while aligning the channel chart to the real coordinates.

Acknowledgment

The authors would like to thank M. Kasparek, A. Eidloth, J. Niklas Bauer, and M. Soliman for implementing the proof-of-concept 5G uplink-TDoA localization setup and its software-defined-radio-based processing pipeline.

References

[1] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3607–3644, 2018.

[2] N. Saeed, H. Nam, T. Y. Al-Naffouri, and M.-S. Alouini, "A State-of-the-Art survey on multidimensional scaling-based localization techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3565–3583, 2019.

[3] Y. Wu, F. Tang, and H. Li, "Image-based camera localization: An overview," *Vis. Comput. Ind., Biomed., Art*, vol. 1, no. 8, pp. 1–13, 2018.

[4] W. Gifford, D. Dardari, and M. Win, "The impact of multipath information on time-of-arrival estimation," *IEEE Trans. Signal Process.*, vol. 70, pp. 31–46, 2020.

[5] M. Stahlke, S. Kram, C. Mutschler, and T. Mahr, "NLOS detection using UWB channel impulse responses and convolutional neural networks," in *Proc. Int. Conf. Localization GNSS*, Tampere, Finland, 2020.

[6] M. Stahlke, S. Kram, F. Ott, T. Feigl, and C. Mutschler, "Estimating TOA reliability with variational autoencoders," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5133–5140, Mar. 2021.

[7] A. Niitsoo, T. Edelhäußer, E. Eberlein, N. Hadaschik, and C. Mutschler, "A deep learning approach to position estimation from channel impulse responses," *Sensors*, vol. 19, no. 5, 2019, Art. no. 1064.

[8] M. Stahlke, T. Feigl, M. H. C. García, R. A. Stirling-Gallacher, J. Seitz, and C. Mutschler, "Transfer learning to adapt 5G AI-based fingerprint localization across environments," in *Proc. IEEE 95th Veh. Technol. Conf.*, Helsinki, Finland 2022.

[9] M. Stahlke, T. Feigl, S. Kram, B. Eskofier, and C. Mutschler, "Uncertainty-based fingerprinting model selection for radio localization," in *Proc. 2023 IEEE 13th Int. Conf. Indoor Positioning Indoor Navigation*, Nuremberg, Germany, 2023.

[10] M. Widmaier, M. Arnold, S. Dorner, S. Cammerer, and S. ten Brink, "Towards practical indoor positioning based on massive MIMO systems," in *Proc. IEEE 90th Veh. Technol. Conf.*, Honolulu, HI, USA, 2019.

[11] C. Studer, S. Medjkouh, E. Gonultaş, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47682–47698, 2018.

[12] M. Stahlke, G. Yammine, T. Feigl, B. M. Eskofier, and C. Mutschler, "Indoor localization with robust global channel charting: A time-distance-based approach," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, no. 1 pp. 3–17, Mar. 2023.

[13] T. Yassine, L. L. Magoarou, M. Crussière, and S. Paquelet, "Optimizing multicarrier multiantenna systems for LoS channel charting," *IEEE Trans. Wireless Commun.*, early access, Jun. 28, 2024, doi: 10.1109/TWC.2024.3417892.

[14] P. Stephan, F. Euchner, and S. T. Brink, "Angle-delay profile-based and timestamp-aided dissimilarity metrics for channel charting," *IEEE Trans. Commun.*, early access, Apr. 9, 2024, doi: 10.1109/TCOMM.2024.3386571.

[15] L Le Magoarou, "Efficient channel charting via phase-insensitive distance computation," *IEEE Wireless Commun. Lett.*, vol. 10, no. 12, pp. 2634–2638, Dec. 2021.

[16] P. Agostini, Z. Utkovski, and S. Stańczak, "Channel charting: An Euclidean distance matrix completion perspective," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 5010–5014.

[17] F. Euchner, P. Stephan, M. Gauger, S. Dörner, and S. T. Brink, "Improving triplet-based channel charting on distributed massive MIMO measurements," in *Proc. IEEE 23rd Int. Workshop Signal Process. Adv. Wireless Commun.*, Oulu, Finland, 2022.

[18] H. Al-Tous, O. Tirkkonen, and J. Liang, "Adaptive sector splitting based on channel charting in massive MIMO cellular systems," in *Proc. IEEE 93rd Veh. Technol. Conf.*, Helsinki, Finland, 2021.

[19] H. Al-Tous, T. Ponnada, C. Studer, and O. Tirkkonen, "Multipoint channel charting-based radio resource management for V2V commun," *EURASIP J. Wireless Commun. Netw.*, 2020, Art. no. 132.

[20] T. Ponnada, H. Al-Tous, and O. Tirkkonen, "Location-free beam prediction in mmWave systems," in *Proc. IEEE 93rd Veh. Technol. Conf.*, Helsinki, Finland, 2021.

[21] S. Taner, V. Palhares, and C. Studer, "Channel charting in real-world coordinates," in *Proc. IEEE Glob. Commun. Conf.*, 2023, pp. 3940–3946.

[22] E. Lei, O. Castañeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese neural networks for wireless positioning and channel charting," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput.*, 2019, pp. 200–207.

[23] P. Ferrand, A. Decurninge, L. G. Ordoñez, and M. Guillaud, "Triplet-based wireless channel charting: Architecture and experiments," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2361–2373, Aug. 2021.

[24] J. Deng, O. Tirkkonen, J. Zhang, X. Jiao, and C. Studer, "Network-side localization via semi-supervised multi-point channel charting," in *Proc. Int. Wireless Commun. Mobile Comput.*, 2021, pp. 1654–1660.

[25] Q. Zhang and W. Saad, "Semi-supervised learning for channel charting-aided IoT localization in millimeter wave networks," in *Proc. IEEE Glob. Commun. Conf.*, Madrid, Spain, 2021.

[26] J. Deng, S. Medjkouh, N. Malm, O. Tirkkonen, and C. Studer, "Multipoint channel charting for wireless networks," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, 2018, pp. 286–290.

[27] P. Ferrand, A. Decurninge, L. G. Ordoñez, and M. Guillaud, "Triplet-based wireless channel charting," in *Proc. IEEE Glob. Commun. Conf.*, Taipei, Taiwan, 2020.

[28] B. Rappaport, E. Gönültaş, J. Hoydis, M. Arnold, P.K. Srinath, and C. Studer, "Improving channel charting using a split triplet loss and an inertial regularizer," in *Proc. 17th Int. Symp. Wireless Commun. Syst.*, Berlin, Germany, 2021.

[29] F. G. Zanjani et al., "Modality-agnostic topology aware localization," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 10457–10468, 2021.

[30] T. Ponnada, H. Al-Tous, O. Tirkkonen, and C. Studer, "An out-of-sample extension for wireless multipoint channel charting," in *Proc. Int. Conf. Cogn. Radio Oriented Wireless Netw.*, 2019, pp. 208–217.

[31] C. Geng, H. Huang, and J. Langerman, "Multipoint channel charting with multiple-input multiple-output convolutional autoencoder," in *Proc. IEEE/ION Position, Location Navig. Symp.*, 2020, pp. 1022–1028.

[32] P. Huang et al., "Improving channel charting with representation-constrained autoencoders," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun.*, Cannes, France, 2019.

[33] T. Yassine, L. Le Magoarou, S. Paquelet, and M. Crussière, "Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting," in *Proc. 23rd Int. Workshop Signal Process. Adv. Wireless Commun.*, Oulu, Finland, 2022.

[34] K. S. Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *Proc. Int. Conf. Robot. Automat.*, vol. 4, 1997, pp. 2783–2788.

[35] X. Hou and J. Bergmann, "Pedestrian dead reckoning with wearable sensors: A systematic review," *IEEE Sensors J.*, vol. 21, no. 1, pp. 143–152, 2020.

[36] D. Chicco, "Siamese neural networks: An overview," in *Artificial Neural Networks*. New York, NY, USA: Humana, 2021, pp. 73–94.

[37] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, 2018.

[38] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18066–18074, 2017.

[39] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.

[40] F. Euchner, M. Gauger, S. Dörner, and S. ten Brink, "A distributed massive MIMO channel sounder for "big CSI data"-driven machine learning," in *Proc. 25th Int. ITG Workshop Smart Antennas*, French Riviera, France, 2021.

[41] F. Euchner and M. Gauger, "CSI Dataset dichasus-cf0x: Distributed antenna setup in industrial environment, Day 1," DaRUS, V2, 2022, doi: 10.18419/darus-2854.

[42] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," in *Proc. IEEE Int. Symp. Intell. Signal Process.*, 2009, pp. 37–42.