Implicit Channel Charting with Application to UAV-aided Localization

Pham Q. Viet and Daniel Romero

Dept. of Information and Communication Technology, University of Agder, Grimstad, Norway. Email:{viet.q.pham,daniel.romero}@uia.no.

Abstract—Traditional localization algorithms based on features such as time difference of arrival are impaired by non-line of sight propagation, which negatively affects the consistency that they expect among distance estimates. Instead, fingerprinting localization is robust to these propagation conditions but requires the costly collection of large data sets. To alleviate these limitations, the present paper capitalizes on the recently-proposed notion of channel charting to learn the geometry of the space that contains the channel state information (CSI) measurements collected by the nodes to be localized. The proposed algorithm utilizes a deep neural network that learns distances between pairs of nodes using their measured CSI. Unlike standard channel charting approaches, this algorithm directly works with the physical geometry and therefore only implicitly learns the geometry of the radio domain. Simulation results demonstrate that the proposed algorithm outperforms its competitors and allows accurate localization in emergency scenarios using an unmanned aerial vehicle.

Index Terms—Channel charting, UAV-assisted localization.

I. Introduction

Localization services play a central role in countless applications such as navigation, augmented reality, autonomous driving, wireless communications and emergency response to name a few. Most localization systems rely on algorithms that provide location estimates based on pilot signals that are received from satellites or terrestrial transmitters. In case of line-of-sight (LOS) reception, modelbased approaches are typically pursued, where geometric principles are applied to estimate locations from distance and/or angle estimates obtained from channel features such as time of arrival, time difference of arrival, or angle of arrival. In turn, when there is not LOS to a sufficient number of transmitters, as occurs indoors or in urban scenarios, data-driven approaches are preferred since the aforementioned distance or angle estimates become too inaccurate. The most prominent example of this class of algorithms is fingerprinting, which involves recording a set of channel state information (CSI) vectors measured at known locations; see [1] and references therein. Location estimates can be obtained, for instance, by comparing the CSI observed by the node to be located with the entries of this data set and applying K-nearest neighbors. More

This work has been funded by the IKTPLUSS grant 311994 of the Research Council of Norway.

sophisticated alternatives rely on deep neural networks (DNNs) to learn a mapping from CSI [2], [3] or from preprocessed CSI [4]–[6] into location estimates. The main limitation of fingerprinting approaches stems from the need for large data sets, which are costly to acquire since each entry involves obtaining the position of a sensor either manually or by means of auxiliary localization systems, e.g. by using a robot.

To alleviate the cost of data collection, channel charting [7] has been recently proposed. The idea is to establish a connection between the geometry in the radio space where (features of) the CSI vectors reside and the geographical geometry of the physical space where the nodes to be located lie. The key assumption is that CSI vectors acquired at spatially near locations are similar to each other. Fig. 1a depicts the main steps in channel charting. There, a dimensionality reduction algorithm assigns a point in 2D or 3D space to each input CSI vector in such a way that the distance between each pair of points is similar in some sense to the dissimilarity between the feature representations of the CSI vectors acquired at those points; see Sec. II. This mapping is referred to as a *channel chart*. The relative positions of the points it returns approximately correspond to the relative positions of the nodes in the physical space. If in addition there are enough anchor nodes, i.e. nodes whose positions are known, semi-supervised extensions [8] can provide absolute position estimates.

In early works on channel charting, feature extraction and dissimilarity metrics are manually engineered by relying on physical principles and heuristic considerations. To reduce the inaccuracies arising from these approaches, DNN-based alternatives learn one of these steps from data. For instance, [7] and [9] fix the feature extraction step and learn the dissimilarity metric or correspondence between the CSI vectors and the channel chart. Conversely, [10] learns the mapping from CSI to features while fixing the dissimilarity metric to be the Euclidean distance. In short, both approaches learn only part of the workflow. Besides, the explicit construction of a channel chart is convenient in those applications where only relative positions are required, but bypassing such a step is naturally expected to result in improved localization performance when absolute positions are needed.

Building upon these two observations, the present

work proposes implicit channel charting-based localization (ICCL), where the radio geometry is learned from data without explicitly constructing a channel chart. In the first step, a DNN is used to predict the physical (or geographical) distances between nodes given the CSI that they measure. In the second step, these distances are utilized in combination with the locations of anchor nodes to estimate the absolute positions of the nodes. Thus, unlike most channel charting schemes, ICCL is supervised and provides absolute location estimates. Relative to modelbased localization algorithms, the proposed scheme inherits the robustness of fingerprinting to non-LOS (NLOS) propagation. As compared to conventional fingerprinting, the proposed algorithm learns the radio geometry from data, whereas relative to DNN-based fingerprinting, learning is heavily improved since the fact that distances are learned instead of absolute positions gives rise to a natural data augmentation effect, where the number of training examples is quadratic in the number of entries of the data set; cf. Sec. IV-B2. Finally, the proposed scheme leverages CSI acquired by multiple nodes rather than from only one, which is expected to increase robustness to noise and reduce the size of the required data set.

Although the proposed ICCL approach could be used with arbitrary forms of CSI, this paper focuses on a scenario where an unmanned aerial vehicle (UAV) is used to locate nodes on the ground. This is well motivated when no terrestrial infrastructure is operational because of a natural disaster or a military attack and when no global navigation satellite systems (GNSSs) can be used, e.g. because nodes lack the appropriate sensors or because the propagation environment precludes LOS propagation from the satellites. This constitutes another contribution of the paper since, to the best of our knowledge, (i) existing schemes for localization with UAVs rely on model-based algorithms and therefore are sensitive to NLOS conditions and other channel impairments, and (ii) no previous work has considered channel charting in setups involving UAVs.

This paper is organized as follows. After reviewing some relevant background in Sec. II, Sec. III formulates the problem. ICCL is proposed next in Sec. IV and its performance is empirically assessed in Sec. V. Finally, Sec. VI concludes the paper.

Notation. Lower and uppercase boldface letters denote column vectors and matrices, respectively. I denotes the identity matrix of appropriate size. The conjugate transpose operator is $(.)^H$. A circularly-symmetric complex Gaussian distribution with mean μ and variance σ^2 is represented as $\mathcal{CN}\left(\mu,\sigma^2\right)$. Finally, $\|.\|$ denotes the Euclidean norm.

II. CHANNEL CHARTING

Channel charting was proposed in [7] as an *unsupervised* alternative to algorithms such as fingerprinting, which suffer from high data acquisition costs. In this context, *supervised* means that each entry of the data set is a pair of a CSI vector and the location at which it was acquired,

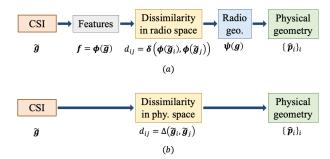


Fig. 1: (a): Conventional (explicit) channel charting. (b): Implicity channel charting (proposed).

whereas *unsupervised* means that each entry of the data set contains just a CSI vector. The price to be paid is that plain channel charting just provides coarse information of the relative locations of the nodes. In some applications, this kind of information suffices to enhance network functionalities such as handover management, predictive radio resource allocation, and user tracking or pairing [9].

As indicated earlier, the core idea behind channel charting is that spatially close sensors are expected to measure similar CSI from the relevant transmitters. To apply this principle, the key steps of channel charting are described next and summarized in Fig. 1a. Consider M nodes located at positions $\{p_i\}_{i=1}^M\subset\mathbb{R}^D$, where D equals 2 or 3. First, the CSI vector $\tilde{\boldsymbol{g}}_i\in\mathbb{C}^L$ acquired by the i-th node is mapped into a feature vector $f_i = \phi(\tilde{g}_i) \in \mathbb{C}^{L'}$. For example, such a transformation may involve computing second-order moments, scaling, and transforming the result into the angular domain [7]. For each pair of nodes, say (i,j), a dissimilarity metric $d_{i,j} = \delta(\phi(\tilde{g}_i), \phi(\tilde{g}_j))$ is subsequently computed. Ideally, function δ should be chosen so that its returned value resembles the physical distance between the locations of these nodes as much as possible. However, this is not generally doable and, for example, [11] uses the so-called correlation matrix distance whereas [7] uses Euclidean distance. In the next stage, a dimensionality reduction algorithm is applied to find Mpoints $\{z_i\}_{i=1}^M\subset\mathbb{R}^D$ in such a way that the distance between the *i*-th and the *j*-th point is ideally $d_{i,j}$ for all i, j. Sammon's mapping [12] can be used to this end, but other methods such as principal component analysis (PCA) [13] and autoencoders have also been considered [7].

The mapping from \tilde{g}_i to z_i constitutes the channel chart. The vectors z_i are named *pseudopositions* because they approximately preserve the *relative* positions of the vectors p_i . For this reason, the quality of a channel chart is typically quantified by ad-hoc metrics such as the *trustworthiness* and *continuity* [14]–[16]. However, it is also possible to obtain absolute location estimates with channel charting by relying on semi-supervised learning [8].

III. PROBLEM FORMULATION

Consider M nodes located at positions $\{p_i\}_{i=1}^{M} \subset \mathbb{R}^D$, where D equals 2 or 3. The positions \mathcal{P}_a

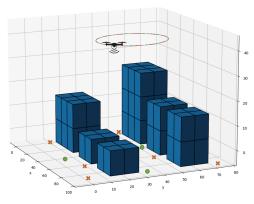


Fig. 2: Illustration of a localization problem in an urban scenario using a UAV. Green circles denote nodes with known locations. Orange crosses represent nodes with unknown locations. Blue blocks denote buildings.

 $\{p_1, p_2, \dots, p_{M_a}\}$ of the first $M_a \geq 3$ nodes are known and, therefore, these nodes are referred to as *anchors*. The locations $\mathcal{P}_u = \{p_{M_a+1}, p_{M_a+2}, \dots, p_M\}$ of the rest of the nodes are unknown and, consequently, these nodes will be referred to as *unknowns*. The unknowns are not able to localize themselves via GNSS, which occurs for example when (i) high buildings obstruct the LOS to satellites, (ii) the nodes are indoors, or (iii) the nodes are covered by debris, as occurs in applications where survivors from an earthquake must be located. The unknowns cannot localize themselves using the terrestrial infrastructure either, which is relevant when the latter is not operational due to a natural disaster, a military attack, or a long blackout.

To localize the unknowns, a UAV flies over the area and transmits pilot signals at N waypoints $\{u_n\}_{n=1}^N\subset\mathbb{R}^3$ along its trajectory. Although this paper considers a single UAV, it is straightforward to accommodate multiple UAVs. For each of these N waypoints, each node measures the CSI as described next. If the application at hand demands that the UAV locates the nodes, then all nodes report their measured CSI to the UAV. If, instead, each unknown must localize itself, the anchors send their measured CSI vectors to the UAV and the latter broadcasts them to all unknowns. The entire setup is illustrated in Fig. 2.

At the n-th waypoint, the UAV transmits a pilot sequence consisting of N_p symbols denoted as $\boldsymbol{x}_n = [x_n[1], x_n[2], \dots, x_n[N_p]]^{\top}$. For simplicity, assume that both the UAV and the nodes have a single antenna and that the channel is neither frequency nor time selective. Therefore, the channel between the n-th waypoint and the i-th node can be represented by a single coefficient $h_{i,n} \in \mathbb{C}$. The signal received at node i is given by

$$\mathbf{y}_{i,n} = h_{i,n} \mathbf{x}_n + \mathbf{w}_{i,n}, \tag{1}$$

where $\boldsymbol{w}_{i,n} = [w_{i,n}[1], \dots, w_{i,n}[N_p]]^{\top}$ models noise.

Given the anchor positions \mathcal{P}_a , the pilot sequences $\{x_n\}_n$, and the received signals at all nodes $\{y_{i,n}\}_{i,n}$, the problem is to estimate the positions \mathcal{P}_u of the unknowns.

IV. IMPLICIT CHANNEL CHARTING-BASED LOCALIZATION

This section proposes ICCL to solve the problem formulated in Sec. III. The algorithm consists of three phases. First, CSI needs to be extracted from the received signals. Given the extracted CSI, a DNN predicts geographical distances between each pair of nodes. Finally, the multilateration algorithm [17] is used to recover the absolute positions of the unknowns given the aforementioned distances and the anchor locations. The key steps in the proposed algorithm are shown in Fig. 1b. Details of each phase will be provided in the following subsections.

A. CSI Extraction

Although ICCL can be applied, in principle, to arbitrary forms of CSI, for concreteness and simplicity, CSI in this paper refers to the power gain.

In view of the model in (1), the least-squares estimator of $h_{i,n}$ given $y_{i,n}$ and x_n is given by

$$\tilde{h}_{i,n} = \boldsymbol{x}_n^H \boldsymbol{y}_{i,n} / (\boldsymbol{x}_n^H \boldsymbol{x}_n). \tag{2}$$

An estimate of the power gain can therefore be obtained as $\tilde{g}_{i,n} = |\tilde{h}_{i,n}|^2$. The CSI vector of the *i*-th node can then be defined as $\tilde{g}_i = [\tilde{g}_{i,1}, \tilde{g}_{i,2}, \dots, \tilde{g}_{i,N}]^{\top} \in \mathbb{R}^N$.

For pre-training purposes, as discussed later, it is convenient to be able to generate samples of $\tilde{g}_{i,n}$ without simulating the propagation of the pilot signals through the channel as per (1). To this end, one can set $h_{i,n} = \sqrt{g_{i,n}}e^{j\varphi_{i,n}}$, where $g_{i,n} \in \mathbb{R}_+$ is the true power gain provided by some model, and $\varphi_{i,n} \sim \mathcal{U}(-\pi,\pi)$. Observe that if $\boldsymbol{w}_{i,n} \sim \mathcal{CN}\left(\mathbf{0},\sigma^2\boldsymbol{I}\right)$, then $\boldsymbol{y}_{i,n} \sim \mathcal{CN}\left(\boldsymbol{x}_nh_{i,n},\sigma^2\boldsymbol{I}\right)$ and, as a result, $\tilde{h}_{i,n} \sim \mathcal{CN}\left(h_{i,n},\sigma^2/\|\boldsymbol{x}_n\|^2\right)$. Then, one could equivalently write $\tilde{h}_{i,n}$ as $\tilde{h}_{i,n} = \sqrt{g_{i,n}}e^{j\varphi_{i,n}} + z_{i,n}$, where $z_{i,n} \sim \mathcal{CN}\left(0,\sigma^2/\|\boldsymbol{x}_n\|^2\right)$ models measurement error. Since the noise is circularly symmetric, one can set $\varphi_{i,n} = 0$ without loss of generality, which yields

$$\tilde{g}_{i,n} = \left| \tilde{h}_{i,n} \right|^2 = \left| \sqrt{g_{i,n}} + z_{i,n} \right|^2.$$
 (3)

Thus, samples of $\tilde{g}_{i,n}$ generated according to (3) are distributed as if the transmission of the pilot signals is simulated through (1) and (2) is evaluated.

B. From CSI to Distances

This subsection presents the process of predicting distances between nodes from their CSI vectors $\{\tilde{g}_i\}_i$. A DNN is trained to this end and, therefore, it will be forced to implicitly learn the geometry in the CSI space.

1) Architecture: Given the CSI vectors \hat{g}_i and \hat{g}_j , the DNN obtains $\Delta_{\theta}(\tilde{g}_i, \tilde{g}_j)$, where θ is a vector collecting all its trainable parameters. This function will be fitted to the distances $\|p_i - p_j\|$. Since $\|p_i - p_j\| = \|p_j - p_i\|$, i.e., the distance from node i to node j equals the distance from node j to node j, the learned function must be invariant to permutations of its inputs, i.e. $\Delta_{\theta}(\tilde{g}_i, \tilde{g}_j) = \Delta_{\theta}(\tilde{g}_j, \tilde{g}_i)$. This could be approximately achieved while training by

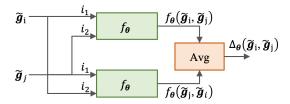


Fig. 3: Architecture of the proposed network.

providing the network with each pair of nodes in both orders, i.e., with the examples $((\tilde{\boldsymbol{g}}_i, \tilde{\boldsymbol{g}}_j), \|\boldsymbol{p}_i - \boldsymbol{p}_j\|)$ and $((\tilde{\boldsymbol{g}}_j, \tilde{\boldsymbol{g}}_i), \|\boldsymbol{p}_i - \boldsymbol{p}_j\|)$ for all i, j. However, a more accurate and efficient approach is to impose invariance by means of the network architecture. To this end, one can let

$$\Delta_{\boldsymbol{\theta}}(\tilde{\boldsymbol{g}}_i, \tilde{\boldsymbol{g}}_j) = \frac{1}{2} \left(f_{\boldsymbol{\theta}} \left(\tilde{\boldsymbol{g}}_i, \tilde{\boldsymbol{g}}_j \right) + f_{\boldsymbol{\theta}} \left(\tilde{\boldsymbol{g}}_j, \tilde{\boldsymbol{g}}_i \right) \right), \tag{4}$$

where f_{θ} is a subnetwork. This is shown in Fig. 3. Observe that, with this architecture, only the pairs of nodes with i < j need to be provided at training time.

For example, the subnetwork f_{θ} used in Sec. V comprises the following layers: convolutional 2D, max pooling, convolutional 2D, max pooling, convolutional 2D, fully connected, and fully connected. Each 2D convolutional layer has 64 filters and 3×2 kernels, except the last one, which has a 3×1 kernel. The pool size of the 2D maxpooling layers is 2×1 . The fully connected layers have 64 and 1 units, respectively.

2) Training Process: Training data can be collected in the same way as for fingerprinting. In the specific setup considered here, the UAV may start operating and sensors equipped with GNSS or other localization systems (e.g. as in LTE or 5G) can be sequentially placed at different positions where they measure the CSI. In case of emergency response applications, this measurement campaign is performed before the natural disaster or military attack.

Once data is acquired, supervised learning is used to train the DNN. The cost function is the mean square error:

$$C(\boldsymbol{\theta}) \propto \sum_{i=1}^{M_0-1} \sum_{j=i+1}^{M_0} \left[\Delta_{\boldsymbol{\theta}} \left(\tilde{\boldsymbol{g}}_i, \tilde{\boldsymbol{g}}_j \right) - \| \boldsymbol{p}_i - \boldsymbol{p}_j \| \right]^2,$$

where M_0 denotes the number of measurement locations in the data set. Observe that the number of training examples is $M_0(M_0-1)/2$, whereas for DNN-based fingerprinting (cf. Sec. I) it would be just M_0 . Thus, the DNN of ICCL is expected to be better trained than the DNN of DNN-based fingerprinting and, as a consequence, the former is expected to outperform the latter.

Nonetheless, DNNs are known to be "data-hungry". Even with M_0 in the order of hundreds, θ may not be learned properly if the network weights are initialized at random. Thus, it is convenient to pre-train the network using another data set, e.g. synthetically generated or measured in a different environment.

C. From Distances to Locations

Given the distance estimates $\hat{d}_{i,j} = \Delta_{\theta}\left(\tilde{\mathbf{g}}_{i}, \tilde{\mathbf{g}}_{j}\right)$ provided by the DNN as well as the anchor locations, ICCL estimates the absolute positions of the unknowns via (possibly iterative) multilateration [17]. The possibility to use this algorithm is a benefit of working directly with physical distances rather than dissimilarity metrics in the radio geometry, as in most channel charting algorithms.

V. EXPERIMENTS

The simulation takes place in an urban area of size 100×80 m. The UAV trajectory is a horizontal circle with center at (40, 45, 40) m and radius 20 m. At N=128 waypoints, he UAV transmits a pilot signal with transmit power $\|\boldsymbol{x}_n\|^2/N_p=30$ dBm. However, the pilot signals are not explicitly generated; cf. Sec. IV-A. The CSI is then measured at $M_0=200$ positions drawn uniformly at random on the ground (D=2). The true power gains $g_{i,n}$ are generated from the 3D city map depicted in Fig. 2 using a tomographic model [18] as in [19]. To focus on impairments in the testing phase, the noise power is set to 0 in the training data but it is greater than 0 for testing data.

The proposed ICCL algorithm is compared with two algorithms. One is the classical distance-based fingerprinting localization (DFPL) algorithm; cf. Sec. I. This algorithm stores the training data. At testing time, given an input CSI vector, this method searches over the stored data and outputs the position that corresponds to the CSI vector that has lowest Euclidean distance to the input. The second algorithm, termed neural-based fingerprinting localization (NFPL), is similar in spirit to those in [2]–[6] but it is applied to the plain CSI vectors introduced in Sec. IV-A. To obtain absolute position estimates, it trains a DNN with the same architecture as the subnetwork of ICCL (cf. Sec. IV-B1) except for minor modifications to accommodate the different input and output size. Specifically, the kernels of the convolutional layers have size of 3×1 instead of 3×2 and the output layer has 2 neurons. Both NFPL and ICCL are pretrained with a data set that comprises $M_0 = 1000$ CSI vectors generated in a different environment, where the buildings have different dimensions.

To quantify the error between the true and estimated locations, the root mean square error RMSE = $\left[\frac{1}{M-M_a}\sum_{j=M_a+1}^{M}\mathbb{E}\left[\|\hat{\boldsymbol{p}}_j-\boldsymbol{p}_j\|^2\right]\right]^{1/2}$ is used, where the expectation runs over realizations of the node locations and measurement noise.

Fig. 4 shows the RMSE of ICCL vs. the number of anchors for different noise levels by averaging over 100 Monte Carlo realizations with M=100 nodes. As expected, the more anchors, the more precise the estimated location. With only 7 anchors, the proposed algorithm can locate unknowns with less than 10-meters average error provided that the noise power is sufficiently low.

Fig. 5 depicts the RMSE of the compared algorithms vs. the noise level by averaging over 100 Monte Carlo realizations with $M_a=20$ anchors and $M-M_a=80$

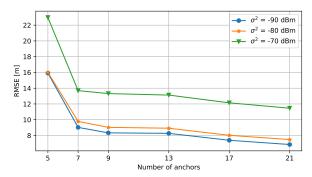


Fig. 4: RMSE of the proposed ICCL algorithm.

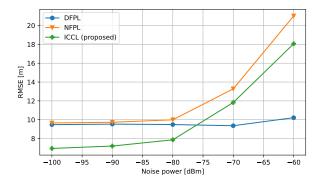


Fig. 5: RMSE of the compared algorithms.

nodes. For a sufficiently small noise level, ICCL outperforms both DFPL and NFPL, which corroborates its ability to learn the radio geometry. However, at large noise power, the accuracy of the ICCL distance estimates degrades and DFPL works better. This is expected to improve if the training data is augmented by adding noise. An apparently counterintuitive fact is that DFPL is seen to outperform NFPL. This phenomenon has already been observed in [1] and may be caused by the fact that DNNs require a large amount of training data. ICCL is less sensitive to this issue, as described in Sec. IV-B2. In contrast, in [6], NFPL offers a better performance than DFPL, but the reason may be that the latter applies a pre-processing step to the CSI vectors. Other works proposing NFPL schemes, such as [2]–[5], do not compare with DFPL.

VI. CONCLUSIONS

This paper proposes implicit channel charting-based localization (ICCL) as a localization approach that implicitly learns the radio geometry of a collection of CSI vectors from a data set. The idea is inspired by channel charting and builds upon the well-known fingerprinting localization method. Simulation results corroborate the merits of the proposed approach.

REFERENCES

- A. Sobehy, É. Renault, and P. Mühlethaler, "CSI-MIMO: Knearest neighbor applied to indoor localization," in *IEEE Int. Conf.* Commun., Jun. 2020, pp. 1–6.
- Commun., Jun. 2020, pp. 1–6.
 M. Arnold, S. Dorner, S. Cammerer, and S. T. Brink, "On deep learning-based massive MIMO indoor user localization," in *IEEE Int. Workshop Signal Process. Advances Wireless Commun.*, Kalamata, Greece, Jun. 2018, IEEE, pp. 1–5.
- [3] M. Arnold, J. Hoydis, and S. Ten Brink, "Novel massive MIMO channel sounding data applied to deep learning-based indoor positioning," in SCC Int. ITG Conf. Syst., Commun. Coding. VDE, Feb. 2019, pp. 1–6.
- [4] A. Niitsoo, T. Edelhäußer, and C. Mutschler, "Convolutional neural networks for position estimation in TDoA-based locating systems," in *Proc. IEEE Int. Conf. Indoor Position. Indoor Navig.* IEEE, Sep. 2018, pp. 1–8.
- [5] X. Li, E. Leitinger, M. Oskarsson, K. Åström, and F. Tufvesson, "Massive MIMO-based localization and mapping exploiting phase information of multipath components," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4254–4267, Sep. 2019.
- [6] P. Ferrand, A. Decurninge, and M. Guillaud, "DNN-based localization from channel estimates: feature design and experimental results," CoRR, vol. abs/2004.00363, Apr. 2020.
- [7] 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, Aug. 2018.
- [8] P. Huang, O. Castañeda, E. Gönültaş, S. Medjkouh, O. Tirkkonen, T. Goldstein, and C. Studer, "Improving channel charting with representation-constrained autoencoders," in *Proc. IEEE Int. Work-shop Signal Process. Advances Wireless Commun.* IEEE, Aug. 2019, pp. 1–5.
- [9] P. Ferrand, A. Decurninge, L. G. Ordoñez, and M. Guillaud, "Triplet-based wireless channel charting," in *IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.
- [10] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Proc. Intl. Conf. Neural Info. Proc. Systems*, vol. 6, 1993.
- [11] P. Agostini, Z. Utkovski, and S. Stańczak, "Channel charting: an euclidean distance matrix completion perspective," in *Proc. - ICASSP IEEE Int. Conf. Acoust. Speech Signal Process.* IEEE, May 2020, pp. 5010–5014.
- [12] J.W. Sammon, "A nonlinear mapping for data structure analysis," IEEE Trans. Comput., vol. C-18, no. 5, pp. 401–409, May 1969.
- [13] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philos. Mag.*, vol. 2, no. 6, pp. 559–572, 1901.
- [14] J. Venna and S. Kaski, "Neighborhood preservation in nonlinear projection methods: an experimental study," in *Artificial Neural Networks*. Springer, 2001, pp. 485–491.
- [15] S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, and E. Castrén, "Trustworthiness and metrics in visualizing similarity of gene expression," *BMC Bioinf.*, vol. 4, no. 1, pp. 1–13, Oct. 2003.
- [16] Á. Vathy-Fogarassy and J. Abonyi, Graph-based clustering and data visualization algorithms, Springer, London, 2013.
- [17] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. ACM MobiCom*, New York, NY, USA, Jul. 2001, MobiCom '01, pp. 166–179, Association for Computing Machinery.
- [18] N. Patwari and P. Agrawal, "NeSh: a joint shadowing model for links in a multi-hop network," in *Proc. IEEE Int. Conf. Acoust.*, Speech, Signal Process., Las Vegas, NV, Mar. 2008, pp. 2873–2876.
- [19] D. Romero, P. Q. Viet, and G. Leus, "Aerial base station placement leveraging radio tomographic maps," in *IEEE Int. Conf. Acoustics Speech Signal Process. Available at arXiv:2109.07372*, Singapore, May 2022.