Enhanced Weighted K-nearest Neighbor Positioning

Xinze Li[1], Hanan Al-Tous[1], Salah Eddine Hajri[2] and Olav Tirkkonen[1]
[1]*Department of Information and Communications Engineering, Aalto University, Finland*
[2]*Huawei Technologies CO. LTD., Lund Research Center, Sweden*
Email: {xinze.li, hanan.al-tous, olav.tirkkonen}@aalto.fi, salah.eddine.hajri@huawei.com

***Abstract*—We consider fingerprinting-based localization in highly cluttered multipath environments with non-line-of-sight conditions, typical of indoor scenarios. Channel state information (CSI) from multiple Base Stations (BSs) is used to construct a fingerprint. We investigate the physical geometry of the k nearest neighbors found by feature distances, as well as possible enhancements to boost achievable positioning accuracy. We observe that the performance of Weighted K-Nearest Neighbor (WKNN) regression depends on the relation between the true position and its k nearest feature neighbors. Better accuracy is achieved when the true position is inside the convex hull of the k nearest neighbors, otherwise localization performance degrades. Consequently, we devise a neighborhood selection algorithm to increase the possibility of a point being inside the convex hull of the k nearest feature neighbors. WKNN localization is also affected by the weighting function used. To further improve performance, we consider a general framework to find the optimum weighting function, utilizing Laguerre polynomials. We benchmark performance against WKNN with exponential weight and deep neural network based localization. Simulation results show that the optimum weighting function with neighbor selection outperforms the benchmark algorithms.**

***Index Terms*—Channel state information, non-line-of-sight communication, fingerprint localization, k-nearest-neighbor regression, Laguerre polynomials.**

I. Introduction

Precise localization is a critical feature of modern wireless networks, owing to a great demand for location-based services and use cases, most notable Industrial Internet-of-Things (IIoT) services. In outdoor scenarios mature localization technologies, such as Global Navigation Satellite System (GNSS), meet the accuracy demands of most services. For indoor scenarios, localization performance is degraded, due to satellite signal attenuation, penetration loss and rich reflections. Fifth Generation New Radio (5G NR) provides enhanced cellular positioning performance, leveraging advanced beamforming capabilities and a wide range of standardized techniques [1], [2].

In challenging indoor localization scenarios, fingerprint localization is a viable technology, with verified performance in many systems [3], [4]. Fingerprint localization is considered as the main direct machine learning-based positioning method in the ongoing 3GPP Rel-18 discussion on Artificial Intelligence (AI) / Machine Learning (ML) for positioning accuracy enhancement. Channel State Information (CSI) features are collected at several known locations to create a database. For example, Received Signal Strength (RSS), Angle of Arrival (AoA), Time of Flight (ToF), Power Delay Profile (PDP) and Channel Impulse Response (CIR) features may be considered. After creating the database, model training is performed, if needed. The location can be estimated directly by Weighted K-Nearest-Neighbor (WKNN) regression, a Support Vector Machine (SVM), or a Deep Neural Network (DNN) [5], [6].

In particular, the WKNN method is widely applied, because of its high accuracy, especially when the data set is small, and due to its low computational cost [7]–[9]. In [8], WKNN shows better localization results than a deep neural network for a measured data set. WKNN performance is affected by feature distance used, e.g. the Euclidean distance, the weight function, e.g. the inverse distance, and the number of neighbors. In [10], the author shows that the Sørensen distance outperforms other distances. In [7], the weight function is modified based on historical trajectory. In [9], the number of neighbors is adjusted adaptively. All these studies focus on RSS feature.

CSI can be used for positioning, as it captures rich information about multi-path propagation channels. To avoid redundant information, CSI data may need pre-processing. In [11], AoAs and ToFs are estimated from CSI to create a database. A two-step k-Nearest-Neighbor (KNN) localization algorithm is designed, where a large set of neighbor candidates is found by ToF and then limited by AoA. In [8], KNN localization based on dimension reduced CSI features is applied.

In this paper, we study the physical geometry of the set of k nearest neighbors found by feature distances. The physical location might be outside of the convex hull of the k nearest neighbors. In such a case, feature neighbors may not be true physical neighbors, so the estimated location based on WKNN is error prone. We consider the spatial covariance matrices from multiple Base Stations (BSs) as a fingerprint, as they capture angle information from the point of view of multiple positions. We apply a log-Euclidean distance to measure fingerprint dissimilarity, since it is a suitable metric for Hermitian positive-definite matrices.

To improve localization performance, we design neighbor selection methods to expand the convex hull with a given number of neighbors. In WKNN localization, we consider a generalized framework to find the optimum weighting function, utilizing Laguerre polynomials. The function is parameterized by a set of parameters obtained by solving an optimization problem aiming to minimize the mean-squared error over the data set. We compare Root-Mean-Squared Error (RMSE) localization performance with a benchmark RSS feature, and with benchmark methods, i.e., WKNN with exponential weight function, and DNN.

The remainder of this paper is organized as follows: In Section II, the system model, and basic concepts are introduced.

In Section III, the convex hull of the k nearest neighbors is analyzed and the neighbor selection methods are introduced. In Section IV, the generalized weight function is introduced. Simulation results are presented and discussed in Section V. Finally, conclusions are drawn in Section VI.

II. System Model

We consider a communication system with B Base Stations (BSs), each BS having M antennas, e.g., a Uniform Linear Array (ULA). User Equipments (UEs) have one omnidirectional antenna. We assume that the UE estimates CSI from downlink reference signals. In an offline phase, the network creates a database, where the fingerprints are channel covariance matrices/received powers for a corresponding physical location. In an online phase, a new fingerprint is measured, and the network estimates the location based on the new fingerprint and the data set.

A. Channel State Information Feature

We assume transmissions based on Orthogonal Frequency-Division Multiplexing (OFDM) with N subcarriers, with a cyclic prefix larger than the maximum delay spread of the channels. The channel vector between UE u and BS b over subcarrier n at time-sample s is $\mathbf{h}_{u,b,n,s} \in \mathbb{C}^{M\times 1}$. The channel coefficients model both path-loss as well as large scale and small scale multipath fading effects. The covariance CSI feature of the channel between UE u and BS b is

$$\mathbf{R}_{u,b} = \frac{1}{S\,N}\sum_{s=0}^{S-1}\sum_{n=0}^{N-1} \mathbf{h}_{u,b,n,s}\,\mathbf{h}_{u,b,n,s}^{\mathrm{H}}\,, \tag{1}$$

where S is the number of time samples and $\mathbf{R}_{u,b} \in \mathbb{C}^{M\times M}$. In $\mathbf{R}_{u,b}$, the effect of small scale fading is averaged which endows covariance-based features with robustness to small scaling fading.

For a fixed transmission power, the received power $s_{u,b}$ is proportional to the average channel gain, $s_{u,b} \sim \mathrm{Tr}\ (\mathbf{R}_{u,b})$. Note that the channel covariance feature $\mathbf{R}_{u,b}$ is smaller in size than the CSI feature$\{\mathbf{h}_{u,b,n,s}\}$ where all frequency and time samples of UE u channels are stacked.

B. Weighted K-Nearest Neighbor Regression

WKNN regression is based on a concept of a local neighborhood. The weight vector is computed based on a feature distance, using a weighting function such as the inverse distance, an exponential function or a Gaussian function.

We denote a CSI-feature by $\mathbf{f}_i$. Let $\mathbf{p}_i$ be the corresponding physical location in the data set, and $d(\circ,\circ)$ the distance measure between two features. To estimate the physical location corresponding to feature $\mathbf{f}_u$, the distances $d(\mathbf{f}_i,\mathbf{f}_u)$ to all points in the data set are computed, and the set $\mathcal{A}_u$ of the k feature points nearest to $\mathbf{f}_u$ are determined, with $|\mathcal{A}_u| = k$. The weight $\omega_{u,i}$ of UE $i \in \mathcal{A}_u$ when localizing UE u is

$$\omega_{u,i} = \frac{g\left(d\big(\mathbf{f}_i,\mathbf{f}_u\big)\right)}{\sum_{i\in\mathcal{A}_u} g\left(d\big(\mathbf{f}_i,\mathbf{f}_u\big)\right)}, \tag{2}$$

where function $g(d)$ maps a distance d to a similarity. The exponential function

$$g_\tau(d) = \exp(-\tau\, d),$$

with tuning parameter τ is the mostly common used function. The location corresponding to feature $\mathbf{f}_u$ is then estimated as:

$$\hat{\mathbf{p}}_u = \sum_{i\in\mathcal{A}_u} \omega_{u,i}\,\mathbf{p}_{u,i}, \tag{3}$$

where $\mathbf{p}_{u,i}$ is the location of neighboring point i in $\mathcal{A}_u$.

C. Feature Distances

The simplest distance between two matrices $\mathbf{M}$ and $\mathbf{M}'$ is the Euclidean distance

$$d_{\mathrm{Euc}}\Big(\mathbf{M},\mathbf{M}'\Big) = \Big\|\mathbf{M}-\mathbf{M}'\Big\|_{\mathrm{F}}. \tag{4}$$

The log-Euclidean distance between two covariance matrices is the Euclidean distance between their matrix logarithms [12]

$$d_{\mathrm{logEuc}}\Big(\mathbf{R},\mathbf{R}'\Big) = \Big\|\log(\mathbf{R}) - \log\Big(\mathbf{R}'\Big)\Big\|_{\mathrm{F}}. \tag{5}$$

It is a metric in the space of positive definite Hermitian matrices, which is a geodesic distance, and both rotation, scale, and inversion invariant [13].

Using the log-Euclidean metric to measure the distance of covariance features can be equivalently understood as considering Euclidean distance between *log-covariance features*. For a positive-definite Hermitian matrix $\mathbf{R}$, the matrix logarithm can be calculated through eigenvalue decomposition:

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}},$$

where $\mathbf{\Lambda}$ is a diagonal matrix. Then

$$\log(\mathbf{R}) = \mathbf{U}\,\mathrm{diag}\left([\log\lambda_1,\ldots,\log\lambda_M]\right)\mathbf{U}^{\mathrm{H}}.$$

For a positive semi-definite matrix, a practical approach is to find the matrix logarithm of the M' largest eigenvalues, with $\lambda_{m'} > 0$ for $m' = 1,\ldots,M'$.

III. Convex Hull of Feature Neighbors

WKNN estimates the location as a linear combination of coordinates of feature neighbors, as in (3). The estimated location is always in the convex hull of the k nearest neighbors. It is important to understand how the feature neighbors in the feature domain are mapped to physical neighbors in the physical domain.

A. Convex Hull Ratio

For this, we investigate whether the true location of a point is in the convex hull of its feature neighbors or not. This can be done by solving a feasibility problem:

$$\mathbf{p}_u - \sum_{i\in\mathcal{A}_u}\alpha_i\,\mathbf{p}_{u,i} = \mathbf{0}, \tag{6a}$$

$$\alpha_i \geq 0,\ i = 1,\ldots,k, \tag{6b}$$

$$\sum_{i\in\mathcal{A}_u}\alpha_i = 1. \tag{6c}$$

Location $\mathbf{p}_u$ is in the convex hull of its feature neighbors, if the above problem has a solution. Note that this counts boundary points as being in the convex hull. We solve problem (6) for a data set of U points, and count the number of points in the convex hull of their feature neighbors, denoted by U_{I}. Then, we define the *convex-hull ratio* as:

$$\eta = \frac{U_{\mathrm{I}}}{U}. \tag{7}$$

This ratio provides a fundamental bound on WKNN localization. If a point is outside the convex hull of its feature neighbors, *no weighting function* can provide accurate localization.

B. Illustrative Example

To understand the performance of WKNN from the perspective of the convex hull ratio, we conduct a simulation experiment. An indoor scenario where 4 BSs with 20 m inter-site-distance are located at the corners of a hall, and 2000 UEs are located on a grid with 0.4 m spacing. Channels are generated using the Quasi Deterministic Radio Channel Generator (QuaDRiGa) simulator [14].

We consider two kinds of CSI-features—covariance and power features as benchmarks. We study the convex-hull ratio performance of log-scale covariance matrix and power constructed from one BS and four BSs. Feature neighbors are found by the Euclidean distance of power, log-power, covariance and log-covariance features.

Figure 1 shows the convex-hull ratio versus the number of feature neighbors k. The covariance and log-covariance has a higher convex-hull ratio than the power and log-power, and the log-scale features show higher convex-hull ratio compared to the linear-scale features. Extending the feature by including more BSs improves the convex-hull ratio, as shown by the convex-hull ratio of four BSs compared to one BSs with log-covariance and log-power feature. The results indicate that the covariance features outperform power features with a wide margin, and similarly log-scale features outperform linear scale features. For this simulated scenario, when $k = 5$, the best performing log-scale covariance feature has 20% of the points outside of the convex hull of feature neighbors, indicating that even with a perfect data driven weighting method, WKNN would result in 20% of the data points erroneously localized.

Figure 2 shows the Cumulative Distribution Function (CDF) of localization error for the points inside and outside the convex hull, based on log-scale benchmark features from four BSs with WKNN using exponential weight function ($\tau = 1$).

The higher the number of neighbors, the higher the convex-hull ratio. The analysis above motivates us to consider a neighbor selection process to improve the convex-hull ratio.

C. Neighbor Selection Method

We devise a neighbor selection method to increase the convex-hull ratio without changing the number of neighbors k, including a neighbor addition rule and a deletion rule.

- Neighbor selection method (NSM) 1: We construct a set $|\mathcal{A}_u|$ of k neighbors for user u, where no three physical locations are near to be on a line, since any point on the line segment is a convex combination of two end points. The set $|\mathcal{A}_u|$ is constructed iteratively, each time adding one point.
- NSM 2: To expand the convex hull we create a set of $k+1$-nearest neighbors. Then calculate the averaged pairwise Euclidean distance of the physical location within each subset of k neighbors and select the one giving the largest pairwise distance.
- NSM 1&2: The set of $k+1$ nearest neighbors is created satisfying NSM 1.

All neighbor selection methods increase convex-hull ratio, as shown in Figure 3. The feature is log-covariance. Figure 4 depicts the selected neighbor coordinates applying rules. The convex hull is expanded after neighbors selection method 2.

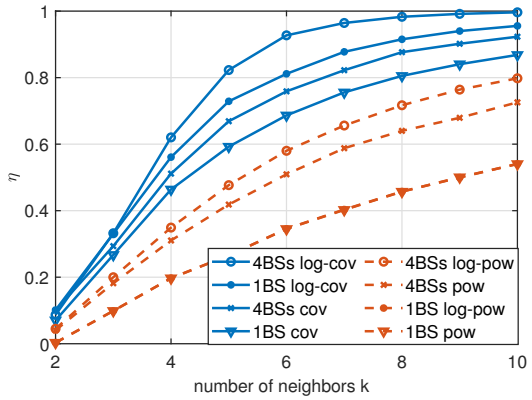

Fig. 1: The convex-hull ratio as a function of the number of neighbors for different CSI features.

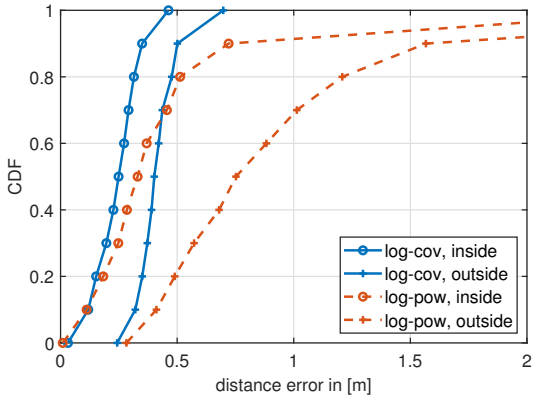

Fig. 2: Cumulative distribution function of localization error for WKNN with exponential weight function based on log-covariance matrices and log-power feature.

IV. Learning the Weight Function

In WKNN regression, the exponential weight function $g_\tau(d)$ with one fixed parameter τ is commonly used. We generalize this to a parameterized weight functions, given as

$$g_{\tau,\mathbf{a}}(d) = e^{-d\tau}\left(a_0 L_0(d) + \ldots + a_m L_m(d)\right), \tag{8}$$

Algorithm 1 Neighbor Selection Method 1

Find neighbor locations $\mathbf{p}_{u,1}, \ldots, \mathbf{p}_{u,U}$ sorted by feature distance, i.e., $d(\mathbf{f}_u, \mathbf{f}_1) < \ldots < d(\mathbf{f}_u, \mathbf{f}_U)$.
Set $\mathcal{A}_u \leftarrow \emptyset$, and $l \leftarrow 0$.
while $|\mathcal{A}_u| < k$ **do**
  $l \leftarrow l + 1$
  if $|\mathcal{A}_u| < 2$ **then**
    $\mathcal{A}_u \leftarrow \mathcal{A}_u \cup \mathbf{p}_{u,l}$
  else
    if $\mathbf{p}_{u,l}$ is not on the line with any two points in $\mathcal{A}_u$ **then**
      $\mathcal{A}_u \leftarrow \mathcal{A}_u \cup \mathbf{p}_{u,l}$
    end if
  end if
end while

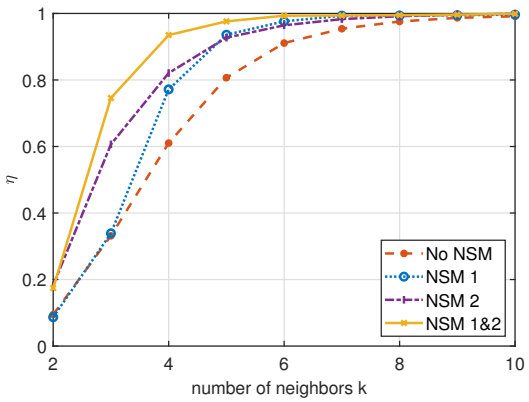

Fig. 3: Convex-hull ratio as a function of the number k of nearest neighbors for different neighbor selection methods. The log-covariance feature is used with 4 BSs.

where the tuning parameters are τ and $\mathbf{a} = [a_0, \ldots, a_m]^{\mathrm{T}}$, and L_m is mth-degree Laguerre polynomial, given as

$$L_m(x) = \sum_{n=0}^{m} \frac{(-1)^n}{n!} \binom{m}{n} x^n.$$

Note that the exponential weight function is a special case of the generalized weight function. The Laguerre polynomials with an exponential kernel are a natural orthogonal basis for functions taking values on the positive real line $\mathbb{R}_+$ [15].

We optimize parameters based on the data set, considering

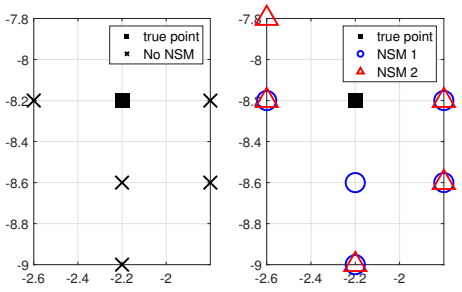

Fig. 4: Examples of selection of $k = 5$ neighbors. Neighbor coordinates shown for different neighbor selection methods.

the sum of squared localization errors as a cost function, i.e.,

$$C(\tau, \mathbf{a}) = \sum_{u=1}^{U} c_u(\tau, \mathbf{a}), \tag{9}$$

where $c_u(\tau, \mathbf{a})$ denotes the squared localization error for point u, given as

$$c_u(\tau, \mathbf{a}) = \left\| \mathbf{p}_u - \frac{\sum_{i=1}^{k} g_{\tau,\mathbf{a}}(d_{u,i})\, \mathbf{p}_{u,i}}{\sum_{i=1}^{k} g_{\tau,\mathbf{a}}(d_{u,i})} \right\|_2^2. \tag{10}$$

The true position for point u is $\mathbf{p}_u = [x_u, y_u]^{\mathrm{T}}$, and correspondingly the neighbor positions are $\mathbf{p}_{u,i}$. The cost function can be re-written as

$$c_u(\tau, \mathbf{a}) = \left\| \mathbf{p}_u - \frac{\mathbf{E}_u(\tau)^{\mathrm{T}}\, \mathbf{M}_u^{\mathrm{T}}\, \mathbf{a}}{\mathbf{e}_u(\tau)^{\mathrm{T}}\, \mathbf{M}_u^{\mathrm{T}}\, \mathbf{a}} \right\|_2^2, \tag{11}$$

where

$$\mathbf{M}_u = \begin{bmatrix} L_0(d_{u,1}) & \ldots & L_0(d_{u,k}) \\ \vdots & \ddots & \vdots \\ L_m(d_{u,1}) & \ldots & L_m(d_{u,k}) \end{bmatrix} \in \mathbb{R}^{(m+1)\times k},$$

the coordinates of the k nearest neighbors are arranged as:

$$\mathbf{P}_u = \begin{bmatrix} x_1, \ldots, x_k \\ y_1, \ldots, y_k \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{k\times 2},$$

while the feature distances to the k neighbors are

$$\mathbf{d}_u = [d_{u,1}, \ldots, d_{u,k}]^{\mathrm{T}} \in \mathbb{R}^{k\times 1},$$

the exponential weight vector is $\mathbf{e}_u(\tau) = e^{-\tau\, \mathbf{d}_u}$, and the matrix

$$\mathbf{E}_u(\tau) = \mathrm{diag}\left(\mathbf{e}_u(\tau)\right)\, \mathbf{P}_u \in \mathbb{R}^{k\times 2}.$$

When the feature distance is 0, we set the weight function to $g_{\tau,\mathbf{a}}(0) = 1$. This results in the constraint $\mathbf{1}^{\mathrm{T}}\mathbf{a} = 1$. The optimization problem then becomes

$$\min_{\tau,\mathbf{a}} C(\tau, \mathbf{a}), \text{ subject to } \mathbf{1}^{\mathrm{T}}\mathbf{a} = 1.$$

This is a non-convex optimization problem. We obtain the optimization parameters by the gradient projection algorithm. The gradient with respect to $\mathbf{a}$ is[1]:

$$\begin{aligned} \frac{\partial c_u}{\partial \mathbf{a}} \;=\; & -2\frac{\mathbf{M}\mathbf{E}\mathbf{p}}{\mathbf{e}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}} + 4\frac{\mathbf{p}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}\mathbf{M}\mathbf{e}}{(\mathbf{e}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a})^2} \\ & + 2\frac{\mathbf{M}\mathbf{E}\mathbf{E}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}}{(\mathbf{e}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a})^2} - 2\frac{\mathbf{a}^{\mathrm{T}}\mathbf{M}\mathbf{E}\mathbf{E}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}\mathbf{M}\mathbf{e}}{(\mathbf{e}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a})^3}, \end{aligned} \tag{12}$$

and the gradient with respect to τ is:

$$\begin{aligned} \frac{\partial c_u}{\partial \tau} \;=\; & -2\frac{\mathbf{p}^{\mathrm{T}}(\nabla_\tau \mathbf{E})^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}}{\mathbf{e}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}} + 4\frac{(\nabla_\tau \mathbf{e})^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}\mathbf{p}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}}{(\mathbf{e}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a})^2} \\ & + \frac{\mathbf{a}^{\mathrm{T}}\mathbf{M}\mathbf{E}(\nabla_\tau \mathbf{E})^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a} + \mathbf{a}^{\mathrm{T}}\mathbf{M}(\nabla_\tau \mathbf{E})\mathbf{E}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}}{(\mathbf{e}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a})^2} \\ & - 2\frac{\mathbf{a}^{\mathrm{T}}\mathbf{M}\mathbf{E}\mathbf{E}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}(\nabla_\tau \mathbf{e})^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a}}{(\mathbf{e}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{a})^3}, \end{aligned} \tag{13}$$

[1]The subscript u for the matrices is ignored, and the parameter τ is removed from $\mathbf{E}(\tau)$ and $\mathbf{e}(\tau)$ in the gradient computations for simplicity.

where $\nabla_\tau \mathbf{e} = -\text{diag}(\mathbf{d})\mathbf{e}(\tau)$, and $\nabla_\tau \mathbf{E} = -\text{diag}(\mathbf{d})\mathbf{E}(\tau)$. Thus the gradients of the cost function are

$$\frac{\partial C}{\partial \mathbf{a}} = \sum_{u=1}^{U} \frac{\partial c_u}{\partial \mathbf{a}}, \qquad \frac{\partial C}{\partial \tau} = \sum_{u=1}^{U} \frac{\partial c_u}{\partial \tau}. \tag{14}$$

In the projected gradient descent, at iteration t we update $\mathbf{a}_t$ and τ_t as follows:

$$\mathbf{a}'_t = \mathbf{a}_{t-1} - \lambda \frac{\partial C}{\partial \mathbf{a}}, \qquad \mathbf{a}_t = \frac{\mathbf{a}'_t}{\mathbf{1}^{\mathrm{T}} \mathbf{a}'_t} \tag{15}$$

$$\tau_t = \tau_{t-1} - \lambda \frac{\partial C}{\partial \tau} \tag{16}$$

where λ is the step size.

V. Simulation

We consider network-based localization, where the UE measures the covariance matrices and reports to the network to create a fingerprint database. The data set is split into a training set and a testing set. For the optimized weight function, the parameters are found based on the training set. The neighbor selection is processed at the network side.

As the covariance is a Hermitian matrix, it can be described in terms of M^2 real numbers for one BS. The log-scale covariance feature is created by first taking the matrix logarithm then taking the real and the imaginary parts to vectorize the matrix. The received power feature is created from the B received powers, consisting of B non-negative values, and the log-scale received power feature is given by the logarithm of the powers. Feature neighbors are found by the Euclidean distances of features summed over all BSs.

For the benchmark localization methods, we consider WKNN with exponential weight function ($\tau = 1$) and DNN. The DNN takes in a vectorized log-covariance and passes it through three fully connected layers of size $[256, 128, 64]$, using the rectified linear unit (ReLU) activation function. The DNN outputs estimated UE coordinates. The Mean Squared Error (MSE) is considered as the loss function.

We evaluate localization performance in a non-line-of-sight environment, specifically an Indoor Factory Sparse Low (InF-SL) scenario of [16]. The simulation parameters are summarized in Table I. The environment layout consists of 4 BSs located at xy-coordinates $[-10,\ 10]$ m, $[-10,\ -10]$ m, $[10,\ 10]$ m and $[10,\ -10]$ m, where 2000 UEs are on a grid with 0.4 m spacing. The basis of evaluation is synthetic channel data generated with the QuaDRiGa simulator, considering large-scale and small-scale effects including multi-path fading [14]. We adopt the values for delay spread, angle-of-arrival and angle-of-departure distributions for the InF-SL scenario discussed in [16]. The log-covariance and log-power features are computed with 50 time samples. The data set is split into 80% and 20% for training and testing, respectively. We study the log-covariance positioning performance based on neighbor selection methods, with generalized weight function in WKNN regression. The Euclidean distance is used to determine the feature neighbors and their distances. The benchmark feature is the received power feature in the dB scale. The features

TABLE I: Simulation Parameters

Parameter	Value	Parameter	Value
Center Freq.	3.5 GHz	Subcarrier Spa.	30 kHz
Scenario	InF-SL	Bandwidth	10 MHz
BS Tx Power	20 dBm	Noise Power	-174 dBm
BS Height	1.5 m	UE Height	1 m
BS Array	1-8 ULA	UE Array	1

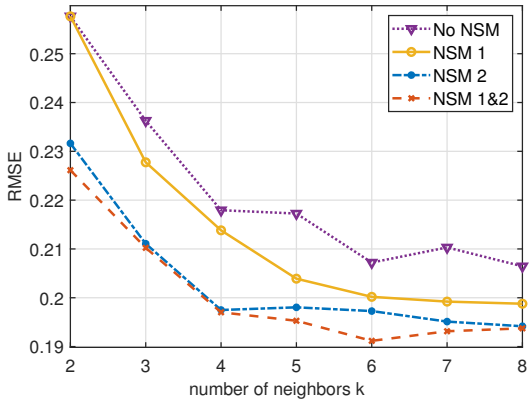

Fig. 5: Localization RMSE against the number of neighbors for neighbor selection algorithms. WKNN localization with exponential weight function ($\tau = 1$).

are in log scale, because the log scale feature shows better localization performance than the linear scale.

First, we investigate neighbor selection methods, including NSM 1, NSM 2 and NSM 1&2. Figure 5 shows RMSE performance with the number of neighbors k. Generally, adding more neighbors improves positioning performance of WKNN regression. The considered neighbor selection methods are beneficial to localization performance. Applying NSM 1&2, the RMSE is decreased to 0.19 m. Table II summarizes the accuracy of these approaches with $k = 5$ neighbors. We use the RMSE, the 80th percentile distance error and 90th percentile distance error as the performance metrics to evaluate and compare the accuracy of the above discussed approaches.

Based on the neighbor selection, we explore the localization performance with polynomial weight functions, considering polynomial degree 2, 3, and 4. For the considered dataset, the RMSE is best for a 3rd degree polynomial, giving an RMSE approximately 45% of the distance between two nearest samples in the data set. Larger polynomial degree leads to overfitting. The 80th percentile distance error for polynomial weight with neighbor selections is 0.33 m, which is smaller than the distance between neighbors on the grid.

Simulation results show that both neighbor selection and learning a polynomial weight function outperforms the benchmark methods. Performance is summarized in Table II. The covariance matrix feature in log scale outperforms the power feature in dB scale. For the covariance feature, both neighbor selection and optimized polynomial weight function are beneficial to localization performance.

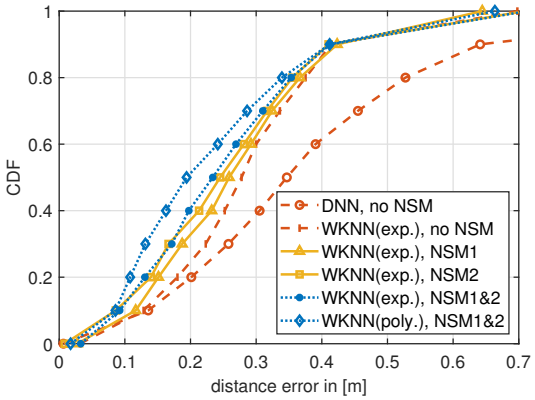

Fig. 6: The CDF of the distance error for log-scale covariance features with $k = 5$ neighbors in WKNN.

TABLE II: Localization Performance of Different Features and Methods with 5 Neighbors

Feature	Method	80%	90%	RMSE
Power	WKNN (exp.)	0.90	1.19	0.57
Covariance	DNN	0.51	0.60	0.29
Covariance	WKNN (exp.)	0.39	0.42	0.22
Covariance	WKNN (poly. 3)	0.37	0.42	0.21
Covariance NS I	WKNN (exp.)	0.37	0.42	0.20
Covariance NS II	WKNN (exp.)	0.36	0.41	0.20
Covariance NS I & II	WKNN (exp.)	0.35	0.41	0.20
Covariance NS I & II	WKNN (poly. 2)	0.35	0.44	0.20
Covariance NS I & II	**WKNN (poly. 3)**	**0.33**	**0.40**	**0.18**
Covariance NS I & II	WKNN (poly. 4)	0.46	0.60	0.45

Figure 6 shows the CDF of the distance error based on WKNN with log-scale covariance matrix feature and compared error distribution applying neighbor selection methods with $k = 5$ neighbors.

VI. Conclusions

In this paper, we have considered WKNN fingerprint-based localization in a challenging NLoS indoor environment. We constructed a CSI fingerprint from log-covariance of several BSs. The neighbors of a test point were found by Euclidean distance. We analyzed geometric characteristics of the set of k-nearest neighbors and the relation to the localization error, observing that localization error increases when the true location is outside the convex hull spanned by the k-nearest neighbors in WKNN. We devised neighbor selection methods aiming to expand the convex hull without increasing the number of neighbors, using physical location information of the neighbors. Specifically, we avoid points lying on a line, and select the neighbor combination to cover a larger area. Furthermore, we considered a generalized weight function whose parameters are optimized based on the data set. The simulation results showed that the RMSE was improved based on the optimum weight function and neighbor selection methods. With optimum weight function and neighbor selections, the RMSE performance was improved by 18% and 38%, compared to the original WKNN using exponential weight and DNN, respectively.

Acknowledgment

This work was funded in part by Huawei Technologies Sweden AB.

References

[1] R. Keating, M. Säily, J. Hulkkonen, and J. Karjalainen, "Overview of positioning in 5G new radio," in *IEEE ISWCS*, 2019, pp. 320–324.
[2] O. Kanhere and T. S. Rappaport, "Position location for futuristic cellular communications: 5G and beyond," *IEEE Commun. Mag.*, vol. 59, no. 1, pp. 70–75, 2021.
[3] P. Bahl and V. Padmanabhan, "RADAR: an in-building rf-based user location and tracking system," in *IEEE Joint Conference of the Computer and Communications Societies*, vol. 2, 2000, pp. 775–784.
[4] P. Castro, P. Chiu, T. Kremenek, and R. Muntz, "A probabilistic room location service for wireless networked environments," in *Int. Conf. Ubiquitous Comput.* Springer, 2001, pp. 18–34.
[5] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, "Indoor intelligent fingerprint-based localization: Principles, approaches and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2634–2657, 2020.
[6] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-Based multi-antenna and multi-point indoor positioning using probability fusion," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2162–2176, 2022.
[7] H. Zhang, Z. Wang, W. Xia, Y. Ni, and H. Zhao, "Weighted adaptive KNN algorithm with historical information fusion for fingerprint positioning," *IEEE Wireless Communications Letters*, vol. 11, no. 5, pp. 1002–1006, 2022.
[8] A. Sobehy, Éric Renault, and P. Muhlethaler, "CSI-MIMO: K-nearest neighbor applied to indoor localization," in *Proc. of ICC*, 2020, pp. 1–6.
[9] S. Liu, R. De Lacerda, and J. Fiorina, "Performance analysis of adaptive k for weighted k-nearest neighbor based indoor positioning," in *VTC2022-Spring*, 2022, pp. 1–5.
[10] J. Torres-Sospedra, R. Montoliu, S. Trilles, Óscar Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems," *Expert Syst. Appl.*, vol. 42, no. 23, pp. 9263–9278, 2015.
[11] Y. Gong and L. Zhang, "Improved k-nearest neighbor algorithm for indoor positioning using 5G channel state information," in *IEEE ITNEC*, vol. 5, 2021, pp. 333–337.
[12] Y. Thanwerdas and X. Pennec, "O(n)-invariant Riemannian metrics on SPD matrices," Sep. 2021.
[13] R. Vemulapalli and D. W. Jacobs, "Riemannian metric learning for symmetric positive definite matrices," *ArXiv*, vol. abs/1501.02393, 2015.
[14] S. Jaeckel, L. Raschkowski, K. Borner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.
[15] M. Abramowitz and I. Stegun., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover Publications, 1964.
[16] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Technical Specification TS 38.901, Jan. 2018, version 14.3.0.