IEEE Sensors Letters VOL. 8, NO. 9, SEPTEMBER 2024 6011404 IEEE Sensors Council

Sensor applications

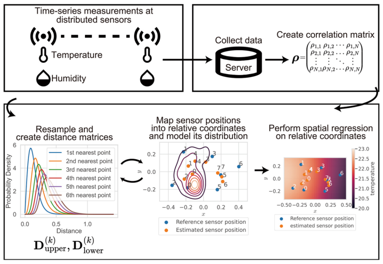
A Location-Blind Spatial Regression Framework for IoT Monitoring Systems Based on Location Distribution and Spatial Correlation

Koki Kanzaki* and Koya Sato**

Artificial Intelligence eXploration Research Center, The University of Electro-Communications, Chofu, Tokyo 182-8585, Japan
**Graduate Student Member, IEEE*
***Member, IEEE*

Manuscript received 29 April 2024; revised 26 July 2024; accepted 6 August 2024. Date of publication 9 August 2024; date of current version 27 August 2024.

Abstract—This letter presents a spatial regression framework that does not rely on absolute positions, such as those obtained from the Global Navigation Satellite System. Typical spatial analysis methods, which depend on precise sensor location data, can result in increased sensor costs and reduced accuracy in challenging environments, such as indoor or underwater settings. Our framework circumvents the need for positioning functions at the sensors by estimating the locational relationships among sensors on relative coordinates based on the probability distribution of the sensor locations and spatial correlations of the sensed data. The server then performs spatial analysis on the relative coordinate system using a regression method, such as Gaussian process regression (GPR) or inverse distance weighting. We validate our approach with two open datasets: a meteorological dataset from the Japanese Meteorological Agency and Intel Lab Data. In both datasets, our results demonstrate that the proposed method can realize spatial regression analysis with less than 10% accuracy degradation in terms of median root mean squared error compared to GPR on absolute coordinates.


Time-series measurements at distributed sensors
Temperature
Humidity
Collect data
Server
Create correlation matrix
$\boldsymbol{\rho}=\begin{pmatrix}\rho_{1,1} & \rho_{1,2} & \cdots & \rho_{1,N}\\ \rho_{2,1} & \rho_{2,2} & \cdots & \rho_{2,N}\\ \vdots & \vdots & \ddots & \vdots\\ \rho_{N,1} & \rho_{N,2} & \cdots & \rho_{N,N}\end{pmatrix}$
Resample and create distance matrices
Probability Density
Distance
1st nearest point
2nd nearest point
3rd nearest point
4th nearest point
5th nearest point
6th nearest point
$\mathbf{D}^{(k)}_{\text{upper}}, \mathbf{D}^{(k)}_{\text{lower}}$
Map sensor positions into relative coordinates and model its distribution
Reference sensor position
Estimated sensor position
Perform spatial regression on relative coordinates
temperature
Reference sensor position
estimated sensor position

Index Terms—Sensor applications, environmental monitoring, Gaussian processes, regression analysis.

I. INTRODUCTION

Environmental monitoring with Internet of Things (IoT) devices has gained significant attention recently [1]. Typical monitoring architectures employ multiple sensors to collect environmental data such as temperature and air quality. This data is then aggregated by a centralized server, which performs regression analysis over spatial or spatio-temporal domains. Environmental information often shows spatial correlation (e.g., [2]). By focusing on the fact that the correlation tends to decrease monotonically with the distance between two observation locations, Gaussian process regression (GPR) [3] or inverse distance weighting (IDW) [4] can interpolate information in unobserved areas [5]. This enables efficient estimation of environmental information from limited observation information and is expected to improve various factors in the society (e.g., power consumption reduction by optimizing air conditioning control and detecting areas dangerous to the human body).

Traditional spatial analysis methods, however, generally rely on precise sensor-location information. Although Global Navigation Satellite System is a common choice for sensor positioning, it often leads to increased sensor costs[1] and significant accuracy degradation in challenging environments, such as indoor or underwater locations [6]. There has been a wide range of discussions aiming to enhance the localization accuracy, such as Angle-of-Arrival (AoA) [7], Time-of-Arrival (ToA) [8], and fingerprinting [9]. Further, recent progress in machine learning enables further localization accuracy: e.g., federated distillation [10], deep learning with radio maps [11], and robust fingerprinting in multifloor indoor environment [12]. However, these approaches require measurements of Wi-Fi signal strength or strict time synchronization between sensors (AoA and ToA), which still does not effectively address the cost issue.

To overcome these challenges, this letter proposes a spatial regression framework that does not rely on absolute positions for distributed monitoring systems. Our approach focuses on visualizing the spatial distribution of environmental data without relying on sensor location information, using a relative coordinate system instead. It begins by estimating the locational relationships among sensors based on the probability distribution of the sensor locations and spatial correlation of the sensed data. The server then performs spatial analysis using these relative coordinates with a regression method. Our results with two open datasets demonstrate that the proposed method can visualize environmental information with accuracy comparable to traditional GPR that uses precise location data.

II. SYSTEM MODEL

We consider an environment where N sensors are distributed in a 2-D area. Each sensor observes time-series data of the interest and records it with the observation time; however, it is not equipped with any positioning function for its location. These sensors are connected to a server by a wireless network, such as a low power wide area network. The server collects data from the sensors and estimates the target information at a specific location and time.

The ith sensor is fixed at an absolute coordinate $\mathbf{x}_i$ in a 2-D space $\mathcal{X}_A$. $\mathbf{x}_i$ is selected based on a probability density function (PDF) $f_l(\mathbf{x})$ (e.g., Poisson point process (PPP) [13]), which is available on the

Corresponding author: Koya Sato (e-mail: k_sato@ieee.org).
Associate Editor: Ashish Pandharipande.
Digital Object Identifier 10.1109/LSENS.2024.3441104

[1] For example, Adafruit Ultimate GPS (https://www.adafruit.com/product/746) costs roughly $30 per module and works at 20 mA operating supply current.

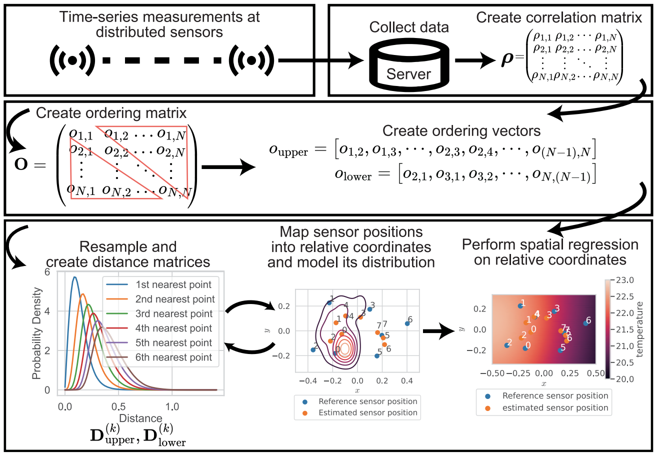

Time-series measurements at distributed sensors
Collect data
Server
Create correlation matrix
$\boldsymbol{\rho}=\begin{pmatrix}\rho_{1,1} & \rho_{1,2} & \cdots & \rho_{1,N}\\ \rho_{2,1} & \rho_{2,2} & \cdots & \rho_{2,N}\\ \vdots & \vdots & \ddots & \vdots\\ \rho_{N,1} & \rho_{N,2} & \cdots & \rho_{N,N}\end{pmatrix}$
Create ordering matrix
$\mathbf{O}=\begin{pmatrix}o_{1,1} & o_{1,2} & \cdots & o_{1,N}\\ o_{2,1} & o_{2,2} & \cdots & o_{2,N}\\ \vdots & \vdots & \ddots & \vdots\\ o_{N,1} & o_{N,2} & \cdots & o_{N,N}\end{pmatrix}$
Create ordering vectors
$o_{\text{upper}} = [o_{1,2}, o_{1,3}, \cdots, o_{2,3}, o_{2,4}, \cdots, o_{(N-1),N}]$
$o_{\text{lower}} = [o_{2,1}, o_{3,1}, o_{3,2}, \cdots, o_{N,(N-1)}]$
Resample and create distance matrices
Probability Density
Distance
1st nearest point
2nd nearest point
3rd nearest point
4th nearest point
5th nearest point
6th nearest point
$\mathbf{D}^{(k)}_{\text{upper}}, \mathbf{D}^{(k)}_{\text{lower}}$
Map sensor positions into relative coordinates and model its distribution
Reference sensor position
Estimated sensor position
Perform spatial regression on relative coordinates
temperature
Reference sensor position
estimated sensor position

Fig. 1. Estimation flow.

server. When the sensors measure the target data at $t = 1, 2, \ldots, T$, the local dataset at the ith sensor can be written as $\mathcal{D}_i = \{[t, y(\mathbf{x}_i, t)] \mid t = 1, 2, \ldots, T\}$, where $y(\mathbf{x}_i, t)$ is the environmental information. The environmental information $y(\mathbf{x}, t)$ varies depending on the coordinate $\mathbf{x}$ and time t. Experimental results in various fields have suggested spatial correlation performance of the environmental information monotonically decreasing with the distance between the sensors (e.g., temperature [2] and received signal power in wireless systems [14]). Based on these empirical discussions regarding the spatio-temporal statistics perspective, we assume that y shows a distance-dependent spatial correlation; i.e., the correlation between two sensing locations, $\mathrm{Cor}[y(\mathbf{x}_i, t), y(\mathbf{x}_j, t)]$, decreases as the distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ increases (e.g., exponential decay model). Further, on the fixed points, it varies based on a stochastic process over the time domain.

The full dataset at the server can be written as $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i$. The server tries to estimate the target value at $\mathbf{x}_*$ and T based on $\mathcal{D}$ and $f_l(\mathbf{x})$. However, it is impossible to estimate $y(\mathbf{x}_*, T)$ on the absolute coordinate system $\mathcal{X}_A$, since the dataset does not contain the sensing locations. Therefore, we herein introduce a 2-D *relative* coordinate area $\mathcal{X}_R$, which is a rectangle normalized to length one on the longest side, and consider mapping an absolute coordinate $\mathbf{x} \in \mathcal{X}_A$ into $\hat{\mathbf{x}} \in \mathcal{X}_R$. The estimation target on the relative coordinates can be expressed as $y(\hat{\mathbf{x}}_*, T)$. The server first estimates $\hat{\mathbf{x}}_i$ in $\mathcal{D}$ based on the correlation of observations and $f_l(\mathbf{x})$. Then, it estimates $y(\hat{\mathbf{x}}_*, T)$ on the relative coordinates via a regression analysis.

III. LOCATION-BLIND SPATIAL REGRESSION

Fig. 1 overviews the proposed framework for estimating $y(\hat{\mathbf{x}}_*, T)$ based on $\mathcal{D}$ and $f_l(\mathbf{x})$. This method focuses on the fact that the spatial correlation of $y(\hat{\mathbf{x}}, t)$ decreases monotonically with the distance between sensors in most spatio-temporal monitoring systems [5]. Our framework estimates the distribution of sensor positions on the relative coordinate by the following three steps: ordering the intersensor distances using the intersensor correlation values, simulating sensor distances based on the ordering information and $f_l(\mathbf{x})$, and performing multidimensional scaling (MDS) [15]. Then, $y(\hat{\mathbf{x}}, t)$ is estimated on $\mathcal{X}_R$ by a regression method, such as GPR.

Let us denote the time-series data vector at the ith sensor as $\mathbf{Y}_i = [y(\mathbf{x}_i, 1), y(\mathbf{x}_i, 2), \ldots, y(\mathbf{x}_i, T)]$. When the correlation coefficient between $\mathbf{Y}_i$ and $\mathbf{Y}_j$ is expressed as $\rho_{i,j}$, the correlation matrix between the sensors can be defined as $\boldsymbol{\rho} (\in \mathbb{R}^{N \times N})$. If the exact correlation model is available on the server, we can predict the relative coordinates using this model and $\boldsymbol{\rho}$. However, such correlation performance depends on the environment, and it is practically impossible to model it when the absolute coordinates of the sensors are unknown. Alternatively, we focus on the monotonically decreasing feature in the spatial correlation and the fact that $f_l(\mathbf{x})$ can be transformed into the distribution of distances to the kth nearest neighbor (NN) sensor. Focusing on the column direction of this matrix, $\rho_{i,j}$ is monotonically decreasing for the Euclidean distance $d_{i,j} \triangleq ||\mathbf{x}_i - \mathbf{x}_j||$. Thus, $\boldsymbol{\rho}$ can be converted into an ordering matrix $\mathbf{O}$, where $o_{i,j}$ represents the order of correlation value with the jth sensor from the ith sensor. This ordering matrix can be divided into upper and lower triangular portions to create two ordering vectors $\mathbf{o}_{\text{upper}}, \mathbf{o}_{\text{lower}} \in \mathbb{R}^{(N(N-1))/2}$; i.e., $\mathbf{o}_{\text{upper}} = [o_{1,2}, o_{1,3}, \ldots, o_{2,3}, o_{2,4}, \ldots, o_{(N-1),N}]$ and $\mathbf{o}_{\text{lower}} = [o_{2,1}, o_{3,1}, o_{3,2}, \ldots, o_{N,(N-1)}]$. The order values in $\mathbf{o}_{\text{upper}}$ and $\mathbf{o}_{\text{lower}}$ are equal to the ones obtained from the exact distance values between the sensors, because f_l can be transformed into a probability distribution f_d representing the kth neighborhood distance, we can estimate the distribution of the sensor position on the relative coordinate by iterating sampling of the distance between sensors and MDS for the distance information. Therefore, we next generate $2L$-distance matrices by sampling from f_d.

The distance matrices created by the lth sampling are denoted by $\mathbf{D}^{(l)}_{\text{upper}}, \mathbf{D}^{(l)}_{\text{lower}} \in \mathbb{R}^{N \times N}$, respectively; after $2L$ matrices are generated, we convert these distance matrices into the sensor locations on relative coordinates based on MDS. In the following, $\mathbf{D}^{(l)}_{\text{upper}}$ and $\mathbf{D}^{(l)}_{\text{lower}}$ can be treated similarly so that they will be referred to together as $\mathbf{D}^{(l)}$ and let $\hat{\mathbf{x}}^{(l)}_i$ be the lth estimated relative coordinate of the ith sensor. The matrix that arranges the relative coordinates can be written as $\hat{\mathbf{X}}^{(l)} = [\hat{\mathbf{x}}^{(l)}_1, \hat{\mathbf{x}}^{(l)}_2, \ldots, \hat{\mathbf{x}}^{(l)}_N]^\top$. The conversion from simulated distance matrix $\mathbf{D}^{(l)}$ to relative coordinate $\hat{\mathbf{X}}^{(l)}$ using MDS is proceeded: MDS first finds the two largest eigenvalues of $-\frac{1}{2}(\mathbf{I} - \frac{1}{N}\mathbf{J})\mathbf{D}^{(l)}(\mathbf{I} - \frac{1}{N}\mathbf{J})$ ($\lambda^{(l)}_1$ and $\lambda^{(l)}_2$) and the corresponding eigenvectors $\mathbf{e}^{(l)}_1, \mathbf{e}^{(l)}_2$ ($\mathbf{J}$ is a $N \times N$ matrix in which all elements are one). Then

$$\hat{\mathbf{X}}^{(l)} = \begin{pmatrix} \mathbf{e}^{(l)}_1 \\ \mathbf{e}^{(l)}_2 \end{pmatrix} \begin{pmatrix} \sqrt{\lambda^{(l)}_1} & 0 \\ 0 & \sqrt{\lambda^{(l)}_2} \end{pmatrix}. \tag{1}$$

Now consider plotting each generated $\hat{\mathbf{x}}^{(l)}$ on $\mathcal{X}_R$. Let $\mathbf{o}$ be the vector of even-rounded averages of the elements of $\mathbf{o}_{\text{upper}}$ and $\mathbf{o}_{\text{lower}}$, and d_i is the mean distance from a sensor to the ith nearest sensor obtained by the distribution f_d. In other words, if $f_d(d, i)$ is a PDF for the distance d from a sensor to the ith nearest sensor, then $d_i = \mathbb{E}[f_d(d, i)]$. Based on d_i, let $\mathbf{d}$ be the vector with the elements of $\mathbf{o}$ replaced by distances. For example, if $\mathbf{o} = [5, 1, 2, 3, 4]$, then $\mathbf{d} = [d_5, d_1, d_2, d_3, d_4]$. A distance matrix $\mathbf{D}$ is created using the vector $\mathbf{d}$. Then, based on $\mathbf{D}$, matrix $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_N]^\top$ can be generated for each sensor position using MDS. Then, based on orthogonal Procrustes analysis [16], each relative coordinate is aligned to $\mathcal{X}_R$ by finding the orthogonal matrix $\Omega^{(l)}$. This matrix can be found by

$$\min_{\Omega^{(l)}} \|\hat{\mathbf{X}}^{(l)}\Omega^{(l)} - \hat{\mathbf{X}}\|, \text{ subject to } \left(\Omega^{(l)}\right)^\top \Omega^{(l)} = \mathbf{I}. \tag{2}$$

Specifically, $\Omega^{(l)}$ in (2) can be found by singular value decomposition (SVD) of $(\hat{\mathbf{X}}^{(l)})^\top\hat{\mathbf{X}}$; if the result of SVD is $\mathbf{U\Sigma V}^*$, $\Omega^{(l)} = \mathbf{UV}^*$. By multiplying $\Omega^{(l)}$ for each $\hat{\mathbf{X}}^{(l)}$ from the right, respectively, all simulated points of the estimated sensor positions can be plotted on $\mathcal{X}_R$. For the plotted points, the distribution of each sensor position on the relative coordinate system $\mathcal{X}_R$ can be estimated by using kernel density estimation [17].

Finally, the target value on $y(\hat{\mathbf{x}}_*, T)$ can be estimated by a regression method on $\mathcal{X}_R$; e.g., GPR and IDW. For example, this regression can be realized by stochastic variational GPR (SVGPR) [18], which is an approximation for GPR. GPR estimates the output in a target point as a random variable following GP. Although the exact GPR can perform an accurate non-parametric regression, it has several drawbacks regarding computational complexity and robustness to uncertain inputs. SVGPR introduces inducing points and variational inference to support large-scale data and uncertain inputs. Assuming that $\hat{\mathbf{Y}} \triangleq \hat{\mathbf{Y}}^{(T)} = [y(\hat{\mathbf{x}}_1, T), y(\hat{\mathbf{x}}_2, T), \ldots, y(\hat{\mathbf{x}}_N, T)]$, and all input points are $\hat{\mathbf{X}}_{\text{all}}$, SVGPR models the predictive distribution of $y_* \triangleq y(\hat{\mathbf{x}}_*, T)$ as $p(y_*|\mathbf{Y}) \approx \mathcal{N}(\mu, \sigma^2)$, where

$$\mu = \mathbf{k}_{\hat{\mathbf{x}}_*\hat{\mathbf{Z}}}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}}^{-1}\sigma_\epsilon^{-2}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}}(\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}} + \sigma_\epsilon^{-2}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{X}}_{\text{all}}}\mathbf{K}_{\hat{\mathbf{X}}_{\text{all}}\hat{\mathbf{Z}}})^{-1}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{X}}_{\text{all}}}\hat{\mathbf{Y}} \tag{3}$$

$$\begin{aligned}\sigma^2 &= k(\hat{\mathbf{x}}_*, \hat{\mathbf{x}}_*) - \mathbf{k}_{\hat{\mathbf{x}}_*\hat{\mathbf{Z}}}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}}^{-1}\mathbf{k}_{\hat{\mathbf{Z}}\hat{\mathbf{x}}_*} \\ &\quad + \mathbf{k}_{\hat{\mathbf{x}}_*\hat{\mathbf{Z}}}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}}^{-1}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}}(\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}} + \sigma_\epsilon^{-2}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{X}}_{\text{all}}}\mathbf{K}_{\hat{\mathbf{X}}_{\text{all}}\hat{\mathbf{Z}}})^{-1}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}}\mathbf{K}_{\hat{\mathbf{Z}}\hat{\mathbf{Z}}}^{-1}\mathbf{k}_{\hat{\mathbf{Z}}\hat{\mathbf{x}}_*}.\end{aligned} \tag{4}$$

Note that σ_ϵ is the noise standard deviation, $\hat{\mathbf{Z}} \in \mathbb{R}^{H \times 2}$ is the inducing points, and H is an integer. We set $H = N$), $k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function, and $\mathbf{K}_{\mathbf{PQ}}$ is the matrix whose each element represents $k(\mathbf{p}_i, \mathbf{q}_i)$.

Further, a pure IDW [4] can be implemented by the following:

$$y_* = \frac{1}{\sum_{i=1}^{N} \frac{1}{\|\hat{\mathbf{x}}_* - \hat{\mathbf{x}}_i\|^w}} \sum_{i=1}^{N} \frac{y(\hat{\mathbf{x}}_i)}{\|\hat{\mathbf{x}}_* - \hat{\mathbf{x}}_i\|^w} \tag{5}$$

where w is a scalar controlling the dependence of weight coefficients on distance (we set $w = 2$)[2].

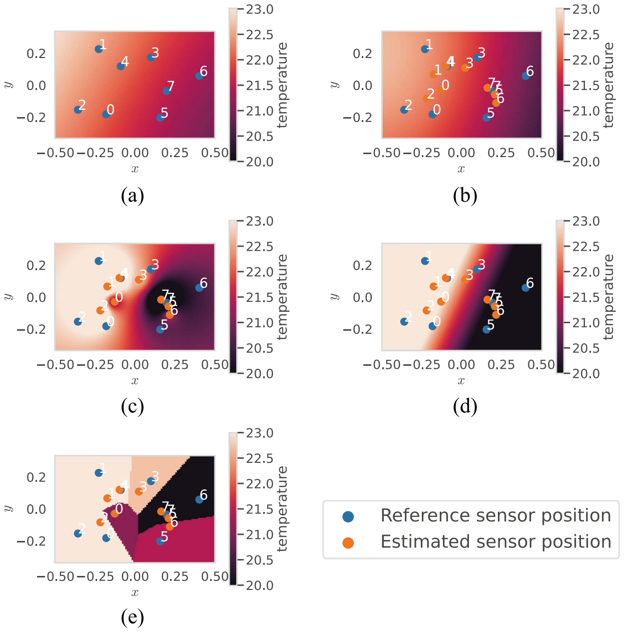


Fig. 2. Example of temperature estimation in Japanese meteorological agency dataset. (a) LGGPR. (b) SVGPR. (c) IDW. (d) LS. (e) NN.

IV. PERFORMANCE EVALUATION

This section evaluates the performances using two open datasets: an outdoor meteorological dataset provided by the Japanese Meteorological Agency[3] and an indoor environmental monitoring dataset in Intel Lab Data.[4] Note that we implemented this simulation on Python 3.12.1 with scikit-learn 1.4.1.post1, scipy 1.12.0, and numpy 1.26.4.

A. Japanese Meteorological Agency Dataset

We extracted 103 meteorological stations that contain a daily temperature, precipitation, wind speed, and sunshine hours for a total of 273 days from January 1 to September 30 in 2023, included in JGD2011/Japan Plane Rectangular CS IX. The positions of these stations were converted to rectangular coordinates; then, we scaled the x- or y-axis, the longer one, to be within the range of [0, 1]. The sensor locations were modeled as a PPP with an average of 18 points per unit area. After candidate points were selected based on PPP, the nearest station from each point was selected as the observation point. We verified the accuracy for 100 randomly-selected datasets.

The accuracy was tested at the unselected sensors on September 30, 2023. To evaluate how the proposed method can approach the location-given method, we selected GPR with exact location information, referred to as *location-given GPR (LGGPR)*, as the baseline. This method performs the GPR for $\mathbf{Y}^{(T)}$ on $\mathcal{X}_A$. Further, in addition to the SVGPR, the proposed method utilized IDW, NN interpolation, and least squares (LS) as the regression method to discuss the effects of the regression method in our framework. Note that GPR-based methods used radial basis function kernel with scaling. Further, f_d was determined empirically by generating one million sets of points based on PPP on the area where the sensor is located. Estimation examples in the temperature monitoring are shown in Fig. 2. Note that, for the demonstration purpose, the longer x- or y-axis was scaled to [0, 1] in LGGPR. The proposed method shows a certain amount of error in mapping sensor positions. However, SVGPR and IDW can estimate the accurate temperature heatmap as in the LGGPR. SVGPR and IDW can estimate the nonlinear and continuous variation of environmental information along the spatial axis through a weighted average that considers spatial correlation with the observed samples. In contrast, LS cannot accurately represent the nonlinearity in spatial variations, often leading to lower precision. Further, NN uses only the nearest sample from the estimation point for inference, disregarding others. While this method maintains the nonlinearity of spatial variations, it can cause discontinuous variations in areas far from samples, resulting in increased error.

Root-mean-squared errors (RMSEs) for temperature [°C], precipitation [mm], wind speed [m/s], and sunshine hours [h], respectively, are shown in Table 1. This table contains the RMSE values of LGGPR and proposed methods at the median (50 percentile) and the 90 percentile performances. To discuss effects of the sensing period on computing the correlation matrix, we introduce a ratio of time-series data used for the inference, r_t $(0 < r_t \leq 1)$. Based on the time indexes in the full data, the server performs the proposed method with data on $t = \lfloor (1 - r_t)T \rfloor + 1, \lfloor (1 - r_t)T \rfloor + 2, \ldots, T$; Table 1 includes the RMSE values under $r_t = 0.3, 0.5, 1.0$. In $r_t = 1.0$ (i.e., inference with full sensing time), the proposed method with SVGPR could estimate an accuracy degradation of about 10% or less than the LGGPR in all observation targets at the median. Further, although the lightweight computation, the IDW was also able to estimate almost as well as SVGPR; in contrast, LS and NN showed less accuracy than SVGPR and IDW. This trend was similar at the 90 percentile. In addition, reducing r_t did not necessarily deteriorate the RMSE performance; i.e., SVGPR or IDW well approximated the LGGPR in various sensing conditions. Spatial correlations of environmental information are not necessarily invariant over time and are affected by climate change and other factors [2]. Therefore, spatial correlation analysis using

[2] This letter implemented the IDW function so that $y_* = y(\hat{\mathbf{x}}_i)$ when a measurement sample satisfies $\|\hat{\mathbf{x}}_* - \hat{\mathbf{x}}_i\| = 0$. Further, a safer implementation for $\|\hat{\mathbf{x}}_* - \hat{\mathbf{x}}_i\| \ll 0$ can be realized if we append a small constant ($\ll 1$) to the distance.

[3] https://www.data.jma.go.jp/gmd/risk/obsdl/

[4] https://db.csail.mit.edu/labdata/labdata.html

Table 1. RMSE in Japanese Meteorological Agency Data

(a)

	Median (50 percentile)					90 percentile				
Target / Method	LGGPR	SVGPR	IDW	LS	NN	LGGPR	SVGPR	IDW	LS	NN
Temperature	2.60	**2.89**	3.03	5.87	4.01	3.19	4.11	**3.71**	12.40	6.11
Precipitation	1.37	**1.47**	2.10	3.45	2.06	3.95	**4.08**	4.25	8.34	5.54
Wind	0.73	0.76	**0.74**	1.11	0.83	0.99	**1.02**	1.03	2.27	1.27
Sunshine hours	1.84	**1.94**	1.95	2.37	2.06	2.35	**2.46**	2.56	5.30	3.29

(b)

	Median (50 percentile)					90 percentile				
Target / Method	LGGPR	SVGPR	IDW	LS	NN	LGGPR	SVGPR	IDW	LS	NN
Temperature	2.60	**2.93**	3.01	6.02	3.50	3.19	4.20	**3.70**	10.54	5.73
Precipitation	1.37	**1.45**	2.12	3.90	2.35	3.95	**4.00**	4.05	14.68	8.84
Wind	0.73	0.76	**0.73**	1.10	0.85	0.99	1.05	**1.02**	2.35	1.26
Sunshine hours	1.84	1.97	**1.94**	2.80	2.12	2.35	**2.46**	2.65	6.05	3.25

(c)

	Median (50 percentile)					90 percentile				
Target / Method	LGGPR	SVGPR	IDW	LS	NN	LGGPR	SVGPR	IDW	LS	NN
Temperature	2.60	**2.95**	3.03	6.21	3.95	3.19	4.16	**3.76**	10.44	6.30
Precipitation	1.37	**1.43**	2.08	3.30	1.86	3.95	**4.05**	4.06	13.71	5.54
Wind	0.73	0.75	**0.73**	1.22	0.90	0.99	1.04	**1.03**	2.55	1.27
Sunshine hours	1.84	1.98	**1.95**	2.56	2.18	2.35	**2.52**	2.60	5.83	3.16

Methods that were closest in accuracy to LGGPR in each category were bolded. (a) r_t= 0.3. (b) r_t = 0.5. (c) r_t= 1.0.

Table 2. RMSE in Intel Lab Data

(a)

	Median (50 percentile)					90 percentile				
Target / Method	LGGPR	SVGPR	IDW	LS	NN	LGGPR	SVGPR	IDW	LS	NN
Temperature	43.7	**44.4**	44.5	66.5	56.5	56.3	**56.3**	58.9	125.5	90.3
Humidity	21.0	21.6	**21.1**	31.4	25.7	26.8	**25.5**	28.0	60.0	43.2
Illuminance	584	**603**	610	887	712	724	**757**	765	1803	950

(b)

	Median (50 percentile)					90 percentile				
Target / Method	LGGPR	SVGPR	IDW	LS	NN	LGGPR	SVGPR	IDW	LS	NN
Temperature	43.7	**44.3**	44.4	70.8	53.1	56.3	**58.5**	59.5	126.1	87.2
Humidity	21.0	21.6	**21.3**	27.5	24.0	26.8	**25.5**	27.8	52.3	38.8
Illuminance	584	**600**	612	801	738	724	**739**	754	1437	1212

(c)

	Median (50 percentile)					90 percentile				
Target / Method	LGGPR	SVGPR	IDW	LS	NN	LGGPR	SVGPR	IDW	LS	NN
Temperature	43.7	**43.7**	43.9	61.9	50.0	56.3	**57.2**	59.4	128.2	82.0
Humidity	21.0	21.7	**21.4**	30.3	24.8	26.8	**24.8**	27.4	66.2	43.0
Illuminance	584	**593**	611	838	714	724	**712**	750	1424	1211

(a) r_t= 0.3. (b) r_t = 0.5. (c) r_t= 1.0.

data over a long time period does not necessarily contribute to RMSE improvement.

B. Intel Lab Data

This dataset contains environmental information collected from 53 sensors deployed at the Intel Barkley Research lab from February 28 to April 5, 2004. These sensors observed temperature, humidity, illuminance, and voltage. Our evaluation selected a total of 29 of these sensors, which output more than 4000 data; in addition, we extracted 349 records in which all 29 sensors observe simultaneously.

RMSE values for temperature [°C], humidity [%], and illuminance [lx] are shown in Table 2. As with the meteorological dataset case, SVGPR and IDW performed well in the indoor environment. This result indicates that even though location-unknown case, the proposed method can estimate with the accuracy close to the location-known method. Overall, for both outdoor/indoor scenarios, SVGPR showed the best accuracy in the proposed method. SVGPR can estimate the mean and variance of the target, although the other methods estimate the mean only. Thus, it will be a better choice for applications requiring the error distribution, such as interference management in wireless communications [19]. Further, IDW also indicated adequate accuracy performance overall. IDW is more computationally efficient than SVGPR because it only computes the inverse of the Euclidean distance. IDW can also be an option for a lightweight analysis.

V. CONCLUSION

We proposed a location-blind spatial regression framework, which works on the relative coordinate system. Evaluation results showed that the accuracy of the proposed method can be sufficiently close to that of the case where the location information is available in both indoor/outdoor environmental monitoring scenarios. Since our method does not rely on absolute coordinate information, it will enable low-cost IoT monitoring systems in a wide range of applications.

ACKNOWLEDGMENT

This work was supported by JST, PRESTO under Grant JPMJPR23P3, Japan.

REFERENCES

[1] A. Salam, *Internet of Things for Sustainable Community Development*. New York, NY, USA: Springer, 2024.

[2] G. J. D. Cecco and T. C. Gouhier, “Increased spatial and temporal autocorrelation of temperature under climate change,” *Sci. Rep.*, vol. 8, no. 1, 2018, Art. no. 14850.

[3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning., Ser. Adaptive Computation and Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[4] D. Shepard, “A two-dimensional interpolation function for irregularly-spaced data,” in *Proc. ACM Nat, Conf., Ser. ACM ’68*. New York, NY, USA, Jan. 1968, pp. 517–524.

[5] N. Cressie and C. K. Wikle, *Statistics for Spatio-Temporal Data*. Hoboken, NJ, USA: Wiley, 2011.

[6] T. Qiu et al., “Underwater Internet of Things in smart ocean: System architecture and open issues,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 7, pp. 4297–4307, Jul. 2020.

[7] T. Margiani et al., “Angle of arrival and centimeter distance estimation on a smart UWB sensor node,” *IEEE Trans. Instrum. Meas.*, vol. 72, Jun. 2023, Art. no. 9508110.

[8] Z. Liu et al., “Machine learning for time-of-arrival estimation with 5 G signals in indoor positioning,” *IEEE Internet Things J.*, vol. 10, no. 11, pp. 9782–9795, Jun. 2023.

[9] Y. Tao et al., “CBWF: A lightweight circular-boundary-based WiFi fingerprinting localization system,” *IEEE Internet Things J.*, vol. 11, no. 7, pp. 11508–11523, Apr. 2024.

[10] Y. Etiabi and E. M. Amhoud, “Federated distillation based indoor localization for IoT networks,” *IEEE Sens. J.*, vol. 24, no. 7, pp. 11678–11692, Apr. 2024.

[11] Ç. Yapar et al., “Real-time outdoor localization using radio maps: A deep learning approach,” *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9703–9717, Dec. 2023.

[12] Y. Zhao et al., “An efficient and robust fingerprint-based localization method for multifloor indoor environment,” *IEEE Internet Things J.*, vol. 11, no. 3, pp. 3927–3941, Feb. 2024.

[13] J. F. C. Kingman, *Poisson Processes*. Oxford, U.K.: Clarendon Press, Dec. 1992.

[14] M. Gudmundson, “Correlation model for shadow fading in mobile radio systems,” *Electron. Lett.*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.

[15] W. S. Torgerson, “Multidimensional scaling: I. theory and method,” *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec. 1952.

[16] P. H. Schönemann, “A generalized solution of the orthogonal procrustes problem,” *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar. 1966.

[17] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Boca Raton, FL, USA: CRC Press, Apr. 1986.

[18] J. Hensman et al., “Scalable variational gaussian process classification,” in *Proc. Artif. Intell. Statost.*, 2015, pp. 351–360.

[19] Y. Zeng et al., “A tutorial on environment-aware communications via channel knowledge map for 6G,” *IEEE Commun. Surveys Tut.*, early access, Feb. 9, 2024, doi: 10.1109/COMST.2024.3364508.