2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Gaussian Gridmaps from Gaussian Processes for WiFi-based Robot Self-Localization in Outdoor Environments

Renato Miyagusuku, Kenta Tabata and Koichi Ozaki

Abstract— **Gaussian Processes have been effectively used to learn location-to-signal-strength mappings from previously acquired observations and enable WiFi-based robot self-localization. However, the cubic computational cost for training and the quadratic cost for prediction with respect to the number of training points limits their scalability, particularly with large datasets necessary for outdoor environments. To reduce prediction cost we propose the use of Gaussian Gridmaps, a spatial representation that stores mean and variance predictions from Gaussian Processes into gridmaps. This approach reduces prediction computational cost to constant time, at the expense of some localization accuracy and increased memory usage. Our experiments demonstrate the feasibility of this method for outdoor localization and examine the impact of quantization and grid resolution on localization performance.**

I. Introduction

Reliable localization is vital for autonomous robot navigation, particularly in GPS-denied environments such as indoor spaces and urban canyons; or crowded areas, where sensor measurements are strongly disrupted by the presence of people or dynamic objects, like range and visual data [1]–[3]. WiFi-based localization has been considered a reliable alternative by utilizing the widespread presence of wireless access points [4]. In this approach, the signal strength from WiFi access points is mapped to create a signal strength map of the environment. A common method involves recording WiFi signal strengths using Received Signal Strength Indicator (RSSI) values, at known locations, creating a "fingerprint" dataset. Then, by matching newly observed signals to the recorded fingerprints, the robot can determine its location within the mapped area, leveraging the distinctive patterns of each location. Matching of new signals can be done by searching the dataset for the most similar signal. Methods such as k-nearest neighbors, random forests, and machine learning, among others, have been used successfully for WiFi-based localization [5]. These methods suffer from decreased localization accuracy when the dataset is sparse, with large gaps between sampled locations. A different approach is to learn location-to-signal-strength mappings, then compute the likelihood of each location being the source of the observed signal. Regression techniques like linear interpolation, smoothing functions, and Gaussian Processes (GPs) have been proposed to model the relationship between spatial locations and WiFi signal strength [6].

Gaussian Processes offer a robust non-parametric framework that models signal strength at any location as a Gaussian distribution, characterized by its mean and variance. This allows GPs to generate both mean predictions and uncertainty estimates, making them well-suited for use with probabilistic algorithms commonly used in robotics. Unfortunately, this capability comes with a high computational cost, GPs have a cubic time complexity relative to the number of data points for training, and a quadratic time complexity for prediction. These computational demands render GPs unsuitable for real-time localization in large-scale environments or with dense datasets.

To mitigate the computational cost of GPs, models can be trained using only a subset of representative data points, named *inducing points* [7]–[9]. By approximating the full GP model using only these inducing points, the computational complexity scales with the number of inducing points rather than the entire dataset, significantly reducing the cost at the cost of complexity - as choosing which points are relevant is not trivial. Another strategy involves dividing the environment into smaller regions, and training separate GP models for each region, thus reducing the dataset size handled by each model. During prediction, only the models relevant to the local region are used, often employing a mixture of experts to smooth transitions at region boundaries [10], [11]. While these approaches may lose some precision or the ability to capture some global trends, they can substantially reduce training and computational costs.

In this work, we propose Gaussian Gridmaps, a spatial representation that stores the mean and variance of WiFi signal strength predictions from Gaussian Processes into a discretized gridmap structure. By pre-computing and storing these predictions, Gaussian Gridmaps reduce the computational cost of prediction to constant time complexity with respect to dataset size, enabling robot localization in any environment. However, because there are typically many more cells than training points, storing Gaussian Gridmaps requires significantly more memory than storing GP models. Additionally, quantization is necessary when storing mean and variance in gridmaps, leading to a tradeoff between increased memory usage and reduced precision due to data quantization. We demonstrate the feasibility of outdoor WiFi localization with real-world experiments, comparing the accuracy of Gaussian Gridmaps against direct GP models. Furthermore, we investigate the impact of different grid cell sizes and quantization levels on localization accuracy

*This work was supported by JSPS KAKENHI Grant Number JP21K14121 and JST Program for co-creating startup ecosystem, Grant Number JPMJSF2319

R. Miyagusuku, K. Tabata and K. Ozaki are with the Graduate School of Engineering, Utsunomiya University, 7-1-2, Yoto, Utsunomiya, Tochigi, Japan {miyagusuku, tabata, ozaki}@cc.utsunomiya-u.ac.jp

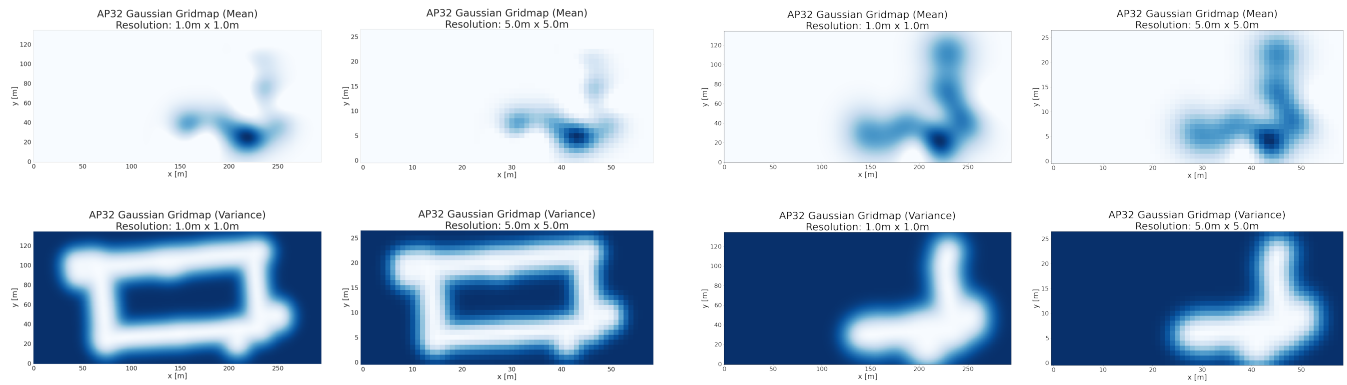

Fig. 1: Example of Gaussian Gridmaps (top) Mean Predictions, (bottom) Variance Predictions at different 1x1 m and 5x5 m resolutions using (left) GP and (right) mGP approach. Darker colors indicate higher values. As GP models set non-observations as 0, the models predict low variances even for unobserved data (top-left area of the map).

and present an efficient sampling method that utilizes this representation for global localization. Although this paper focuses on Gaussian Gridmaps to map wireless signal strength, the approach can be extended to other sensors, such as magnetic sensors, which have also proven useful for robot navigation [9]; as well as to extend approaches that rely on gridmaps, such as occupancy or elevation maps [12], by storing confidence information.

The remainder of this paper is organized as follows. In Section II we introduce Gaussian Gridmaps, including their implementation and an efficient sampling method for global localization. Section III presents experimental results in outdoor environments, along with an ablation study on the effects of grid size and quantization when used as part of the sensor model of a Monte Carlo Localization (MCL) algorithm, as well as for particle sampling to address global localization. Finally, Section IV discusses the findings and concludes the paper with directions for future research.

II. Gaussian Gridmaps

A. Signal Strength Modeling using Gaussian Processes

To generate signal strength maps, our aim is to learn a function $f : \mathbf{x} \rightarrow \mathbf{z}_i$, that predicts signal measurements $\mathbf{z}_i$ for any arbitrary location $\mathbf{x}$. With $\mathbf{z}_i$ being the received signal strength (RSS) from each wireless source (WiFi access points), and $\mathbf{x}$ being x-y Cartesian coordinates. In this work, we do not estimate the heading direction of the robot (θ) nor its effect on signal strength. However, previous work has shown heading direction can be estimated using the angle of arrival of signals - computed using Channel state information (CSI) [13] or antenna arrays [14].

By using Gaussian Processes to learn these mappings, predictions not only provide RSS estimates for any location but also the variance (confidence) of the prediction, which is useful under a probabilistic approach.

Formally, given some training data $(\mathbf{X}, \mathbf{Z})$ where $\mathbf{X} \in \mathbb{R}^{n\times 2}$ is the matrix of n input samples $\mathbf{x}_i, \in \mathbb{R}^2$; and $\mathbf{Z} \in \mathbb{R}^{n\times m}$ the matrix of corresponding output samples $\mathbf{z}_i \in \mathbb{R}^m$. Then, the following assumptions are made:

First, we assume that data pairs $(\mathbf{x}_i, \mathbf{z}_i)$ are drawn from a noisy process:

$$\mathbf{z}_i = f(\mathbf{x}_i) + \epsilon, \tag{1}$$

where ϵ is the noise generated from a Gaussian distribution with known variance σ_n^2

Second, any two output values, $\mathbf{z}_p$ and $\mathbf{z}_q$ are assumed to be correlated by a covariance function based on their input values $\mathbf{x}_p$ and $\mathbf{x}_q$:

$$cov(\mathbf{z}_p, \mathbf{z}_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq}, \tag{2}$$

where $k(\mathbf{x}_p, \mathbf{x}_q)$ is a kernel, σ_n^2 the variance of ϵ and δ_{pq} is one only if $p = q$ and zero otherwise.

Given these assumptions, for any finite number of data points, any individual GP model can be considered to have a multivariate Gaussian distribution:

$$\mathbf{z} \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq}), \tag{3}$$

and therefore be fully defined by a mean function $m(\mathbf{x})$ and a kernel function $k(\mathbf{x}_p, \mathbf{x}_q)$.

For modeling WiFi signal strength (RSS), we first note that RSS is typically measured in decibel-milliwatts (dBm), with values ranging from 0 to -100 (0 dBm being a strong signal and -100 dBm a weak signal). We normalize these values to a range between 1 to 0 (strong to weak signal), assigning 0 for any unobserved signals. Considering these normalized ranges, we choose the zero function as the mean function,

$$m(\mathbf{z}) = 0, \tag{4}$$

and the radial basis function as the kernel function,

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_k^2 \exp\left(-\frac{1}{2}\frac{\|\mathbf{x}_p - \mathbf{x}_q\|^2}{l_k^2}\right). \tag{5}$$

with hyperparameters σ_k^2, also referred to as kernel's variance, and l_k, commonly referred to as lenghtscale.

Mean and kernel hyperparameters θ_g, as well as system noise σ_n are then learned from observed data $(\mathbf{X}, \mathbf{Z})$ by

minimizing the negative log likelihood of $p(\mathbf{Z}|\mathbf{X}, \theta_g, \sigma_n)$,

$$nll = 0.5n\ln(2\pi) + 0.5\ln(\det(\mathbf{K} + \sigma_n^2\mathbf{I}_n)) + 0.5(\mathbf{Z}^T(\mathbf{K} + \sigma_n^2\mathbf{I}_n)^{-1}\mathbf{Z}). \quad (6)$$

Predictions $\mathbf{z}_*$ for an unknown data point $\mathbf{x}_*$, can be made by conditioning $\mathbf{z}_*$ to $\mathbf{x}_*, \mathbf{X}$ and $\mathbf{Z}$, yielding:

$$p(\mathbf{z}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Z}) \sim \mathcal{N}(\mathbb{E}[\mathbf{z}_*], \mathrm{var}(\mathbf{z}_*)), \quad (7)$$

where,

$$\mathbb{E}[\mathbf{z}_*] = \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I}_n)^{-1}\mathbf{z}, \quad (8)$$

$$\mathrm{var}(\mathbf{z}_*) = k_{**} - \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I}_n)^{-1}\mathbf{k}_*, \quad (9)$$

with $\mathbf{K} = cov(\mathbf{X}, \mathbf{X})$ being the $n \times n$ covariance matrix between all training points $\mathbf{X}$, usually called Gram Matrix; $\mathbf{k}_*$ = $cov(\mathbf{X}, \mathbf{x}_*)$ the covariance vector that relates the training points $\mathbf{X}$ and the test point $\mathbf{x}_*$; and k_{**} = $cov(\mathbf{x}_*, \mathbf{x}_*)$ the variance of the test point.

From the selected kernel function, eq. (2), for locations far from the available data in the training dataset $\mathbf{X}$, the covariance vector tends to zero, i.e., if $\|\mathbf{x}_p - \mathbf{x}_q\|^2 \to \infty$, then $\mathbf{k}_* \to 0$. Making predicted values tends the to mean function, which in our case is zero. This is our desired behavior as a zero value aligns with the absence of detectable signals, which is the expected outcome in distant areas. Although unobserved signals are not only the result of signals under the -100 dBm sensing threshold but also result from random errors; in [15] it was observed that for indoor environments, this could be ignored.

Signal strengths outdoors

By assigning a value of zero for all non-observed signals, it is possible to train a single GP model to predict the signal strength of all access points. An alternative approach, preferred in this work, is to learn a different model per access point. Henceforward, we will refer to this approach as mGP. It is important to note that as each model is still a GP, predictions are made using the same eqs. (8, 9).

Learning a model per access point enables the use of different training locations for each access point, avoiding locations where signals were not observed. That is, for each access point j, we create its training set $(\mathbf{X}_j, \mathbf{z}_j)$ with $\mathbf{X}_j \subseteq \mathbf{X}$ being the locations where $\mathbf{z}_j$ is non-zero. Unlike indoor environments, non-observed signals are more frequent in outdoor settings. Due to training sets small sample size, this leads to wrong models. Figure 2 illustrates this issue with an example of learned predicted signals using GP and mGP models. As it can be observed there are several non-observed locations (red 'x' in the right figure) even in-between observed locations. As the model optimizes considering both values equally, predictions tend will be lower than desired. In the mGP case, as these points are not included for predictions, the resulting predicted values more accurately represent signals. The effect of these non-observations could be mitigated by increasing the number of sample points, either by revisiting the same area multiple times, resulting in larger datasets that are more labor-intensive to gather and require models

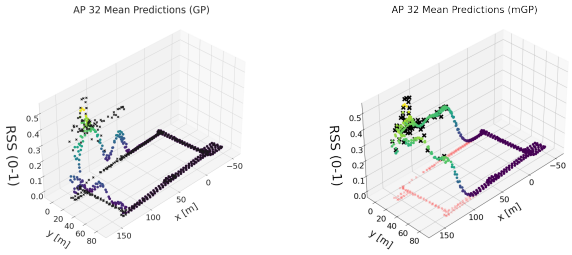

Fig. 2: Example of signal strength predictions using (top) a single GP model and (bottom) a model per access point (mGP). Blue dots represent predicted values while black 'x' shows collected data - for mGP, we also show in red the training points for non-observed signals that are not used.

considerably more compute to train, or by having the robot stop and wait for signals during data collection, significantly extending the time required to gather samples. An alternative approach under exploration is to assign different weights to observed/non-observed signals during optimization using heteroscedastic Gaussian Processes.

Creating gridmaps

To generate the proposed Gaussian Gridmaps, we first partition the environment into a grid composed of $w \times h$ cells c. Formally,

$$G = \{c_{v,u}\} \quad \forall v \in \{0, h-1\}, u \in \{0, w-1\}. \quad (10)$$

In this work, each cell represents an area of equal size $r \times r$ m^2, determined by a resolution parameter r. However, the proposed approach can be applied for mappings with higher dimensions, cells with different lengths per dimension, or variable length sizes.

Considering an environment where signals from m access points can be acquired, we compute the predicted signal strengths (one per access point) and the predicted variance (1 for the GP model and m for the mGP approach) for each cell in the gridmap using eqs. (8, 9); resulting in $m + 1$ or $2m$ gridmaps for the GP and mGP approaches respectively. Figure 1 shows examples of the obtained maps with each model at different resolutions.

Besides map resolution, it is also essential to choose the data format for storing these maps in memory. Common options include unsigned integers (uint8), single precision (float16), or double precision (double).

For the case of unsigned integers, as uint8 ranges from 0-255, we have to scale signal strength predictions and variance predictions. Scaling signal strength predictions is trivial, as the range of RSS signals (0-1) is well defined. For the case of variance predictions, we have to first find the maximum possible variance prediction, which will depend on the model's hyperparameters. In our case, from eqs. (2, 5, 9), we have that for locations far from the available data $\mathrm{var}(\mathbf{z}_*) = k_{**} \to \sigma_n^2 + \sigma_k^2$.

Therefore predicted variances scope is (0 - $\sigma_n^2 + \sigma_k^2$), which we use to scale the values to 0-255. It is important to note

that models with large variances can experience underflow. If the predicted variance is too small, it may get rounded down to zero during quantization because uint8 can't represent very small values accurately. This can cause errors when computing likelihoods (discussed in the next subsection). As a practical solution, we have added code when saving gridmaps so the minimum predicted variance is set to 1 for uint8 quantization.

B. Localization using Gaussian Gridmaps

To realize WiFi-based localization, in this work, we use Monte Carlo Localization (MCL) [16]. MCL is a probabilistic algorithm that estimates the robot's pose using a set of weighted particles to represent possible locations. Given the robot's approximate initial pose, MCL refines the pose estimate through three key steps. First, particles (representing robot poses) are updated based on motion data. Next, particle weights are adjusted using a sensor model that calculates the likelihood of each particle being in the correct pose by comparing new sensor data to predicted sensor values generated based on the particle's pose and a pre-learned map of the environment. Finally, the algorithm re-samples the particles, giving preference to those with higher weights, while discarding lower-weight particles. Repeating these steps enables accurate localization of the robot in the environment.

In previous work [15] a Gaussian Process model was used to compute signal strength predictions and confidence estimations using eqs. (8, 9) directly. The computational complexity for this step is driven by the computation of the $n \times 1$ covariance vector $\mathbf{k}_*$ and its product with the $n \times n$ Gram Matrix $\mathbf{K}$, as other operations like the inverse of the Gram Matrix can be stored in memory in advance. Nonetheless, the matrix-vector multiplication described results in quadratic time complexity, $\mathcal{O}(n^2)$, with respect to the number of training points n. Instead, in this approach we read the pre-computed signal strength and confidence estimates directly from the gridmaps, making the prediction step have linear time complexity, $\mathcal{O}(n)$.

For any point $\mathbf{x}_i$, the cell's index where the information is stored, $c_{v,u}$, are calculated as

$$u = h - \left\lceil \frac{y - o_y}{r} \right\rceil, \quad v = \left\lceil \frac{x - o_x}{r} \right\rceil \tag{11}$$

where h, w are the height and width of the gridmap, r is the grid resolution, and $o_{x,y}$ specifies the $x - y$ coordinates of the left-bottom corner cell.

Using the values retrieved from the Gaussian Gridmaps, we compute the joint likelihood probability of new measurements $\mathbf{z}_*$ conditioned on any arbitrary location $\mathbf{x}_i$ as

$$p(\mathbf{z}_*|\mathbf{x}_*) = \prod_j \mathcal{N}(z_{*,j}|M_j(u,v), V_j(u,v))^{\lambda_j}, \tag{12}$$

$$\lambda_j = \frac{z_{*,j}}{\sum_j (z_{*,j})} \tag{13}$$

where M_j and V_j are the signal strength mean and variance gridmaps, and u, v the cell index correspoding to the arbitrary location $\mathbf{x}_*$ calculated using eq. (11).

Algorithm 1 Sampling using Gaussian GridMaps

```
Compute likelihood gridmap L [eq. (12)]
x_i = x-y location of cells [invert eq. (11)]
w_i = L(u,v)/∑L(u,v) for all cells u, v
c = w[0]
i = 0
r = uniform(0, 1/P)
for p = 0 to P − 1 do
    U = r + (p−1)/P
    while U > c do
        i+ = i
        c+ = w[i]
    end while
    x = uniform(x_i − r/2, x_i + r/2)
    add x to sample set 𝒳
end for
```

Global Localization A more challenging variant of the localization problem is known as the global localization problem, where the robot starts without any prior knowledge of its initial position. Wireless signal strength sensing is particularly well-suited to address this scenario, as it offers some key advantages over other sensing modalities. Due to the large number and widespread distribution of access points throughout most environments, WiFi maps are highly asymmetric - no two places have the same signal strength distribution. Furthermore, each access point broadcasts a unique identifier (MAC address), making the signals easily distinguishable. Considering these factors, WiFi signals offer a fast and reliable way to uniquely determine the robot's location, even in large, repetitive environments, making them an excellent choice for addressing the global localization problem.

In [17], it was proposed to initialize a MCL algorithm by sampling a set of initial particles from the WiFi sensor model. This approach used sampling algorithms to acquire a set of likely particles to initialize a MCL algorithm. Sampling algorithms are used due to the complexity of the sensor model, eq. (12), which makes direct sampling not feasible. Readers are referred to [18] for a detailed explanation of sampling algorithms such as accept-reject, Metropolis-Hastings, and Gibbs, among others. This sampling process can be greatly simplified when using Gaussian Gridmaps by exploiting the ease of sampling discrete vectors of probabilities.

In our proposed approach we first compute the create a set of locations and weights, one for each cell in the Gaussian Gridmaps i.e., $\{c_i, w_i\}^{\forall u,v \in G}$.Then, we compute w as the likelihood map L given new signal strength measurements $\mathbf{z}$, gridmaps M, V, using eq. (12). We sample a set of likely cells using a low variance resampler [16], which is a commonly used discrete sampler in particle filters. For each cell in the new set, we sample a robot pose $p : \{x, y, \theta\}$ considering the coordinate of the cell and its resolution. For any given cell c_i we take sample a location x uniformly at random from $x_i - r/2$ to $x_i + r/2$ where x_i is the global

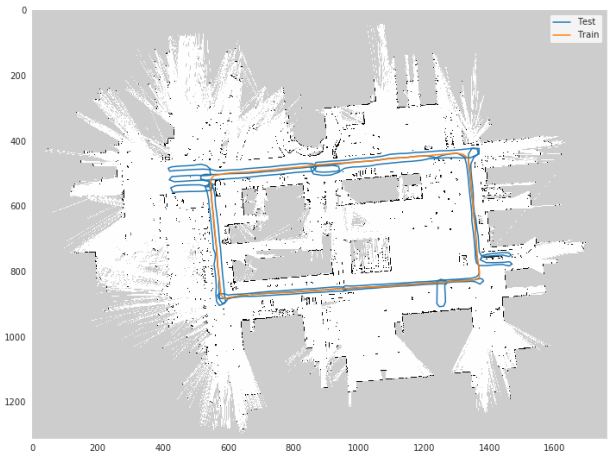

Fig. 3: Map and trajectories taken during experiments: (blue) for creation of training datasets, (orange) for testing.

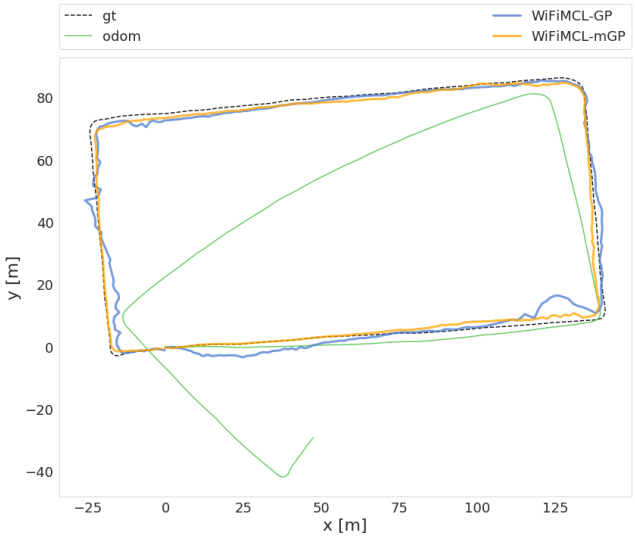

Fig. 4: Results of WiFi-MCL using Gaussian Gridmaps. Dashed lines show the reference trajectory obtained using range data, green shows odometry-only results, and blue and orange show WiFi-MCL results using the GP and mGP approaches respectively.

coordinate associated with c_i and r the resolution of the map. We repeat this process for its y component, while the robot's heading direction θ is sampled at random as L encodes no information regarding the heading direction. Algorithm 1 shows the pseudo-code of our approach.

The cost of this approach is primarily determined by the computation of L. In scenarios with large-scale maps featuring numerous grids, this cost can be substantial. However, by choosing an appropriate map resolution, it is possible to achieve real-time performance for the algorithm.

III. Experimental Results

To validate the proposed method, experiments were carried out at Yoto Campus, Utsunomiya University. Figure 3 shows the map and the trajectory followed by the robot during these experiments, both for creating the maps and for evaluating localization performance. During the experiment, the robot collected WiFi data and range measurements. The range data was used solely to establish ground truth locations for the training datasets and to serve as a reference for assessing localization performance during testing.

A. GP vs mGP

As previously discussed, training a different model per access point (with a different set of training points) allows for better modeling when the likelihood of not observing signals increases. This enhanced modeling considerably affects localization as can be observed in Fig. 4. To account for the stochastic nature of MCL, a MCL using each model (GP, mGP) was run 25 times first computing predictions using eqs. (8,9), and then using our proposed Gaussian Gridmaps. Results are summarized in Table. I

B. Effect of quantization

In order to assess the effect of the quantization of maps, a MCL was run using no quantization (native double precision), unsigned integers using the scaling described in the previous section (uint8), and single precision (float16).

TABLE I: WiFi-MCL errors using GP and mGP models and the Gaussian Gridmaps derived from them

	Direct Predictions		Gaussian Gridmaps	
	GP	mGP	GGM (GP)	GGM (mGP)
Mean	1.99	1.37	2.01	1.40
Std	1.43	0.82	1.41	0.82
Max	8.64	3.47	8.65	3.49

Quantization was done on the mean map, variance map, or both. Results are summarized in Table II.

C. Effect of map resolution

Similar to the previous experiments, in order to assess the effect of map resolution, a MCL was run at several resolutions: 0.1, 0.25, 0.5, 1.0, 2.0, 3.0, and 5.0. Results are summarized in Table III.

IV. Conclusions

Our experiments show the feasibility of employing Gaussian Gridmaps for outdoor WiFi-based localization. To create them, two models were considered, using a single GP to learn the location-to-signal strength for all access points (GP model) or a GP per access point (mGP model). When making predictions directly from the model, the average localization error for the GP model was 1.99m compared to 1.37m when using the mGP model, almost a 30% reduction. Also significant, was the reduction in maximum errors, dropping from 8.64m to 3.47m, a 60% reduction. Although there is a memory penalty as $2m$ maps need to be stored instead of $m+1$, the mGP approach is highly recommended. Notably, when using the proposed Gaussian Gridmaps (at a resolution of 0.1m and with double precision, the localization performance remained almost the same. Regarding the effect of quantization of maps, from Table II it can be concluded that, although

TABLE II: Effect of quantization

Quantization: Mean	double			float16			uint8		
Quantization: Variance	double	float16	uint8	double	float16	uint8	double	float16	uint8
Errors: Mean	1.40	1.41	1.45	1.41	1.41	1.44	1.40	1.41	1.43
Errors: SD	0.82	0.84	0.86	0.85	0.84	0.87	0.83	0.85	0.85
Errors: Max	3.49	3.49	3.74	3.54	6.53	6.64	3.46	3.60	3.62

TABLE III: Effect of map resolution

Resolution	0.1	0.25	0.5	1.0	2.0	3.0	5.0
Mean	1.40	1.45	1.44	1.5	1.83	2.27	3.22
SD	0.82	0.83	0.82	0.84	0.94	1.02	1.38
Max	3.49	3.58	3.47	3.55	3.84	4.28	5.62

there is a slight decrease in performance when using uint8 for variance maps, given an appropriate scaling strategy as the one proposed, the effect is not pronounced. Regarding float16 and double precision, the effect was negligible. Compared to other parameters, map resolution had a more significant impact on localization accuracy. As shown in Table III, the highest resolution of $0.1\text{x}0.1\text{m}^2$ had the same performance as direct predictions from the mGP model; resolutions of 0.25m and 0.5m provided comparable accuracy; and a resolution of 1m resulted in a 7% increase in mean errors, reflecting a slight degradation in performance, though still acceptable. Coarser resolutions, however, showed a more substantial decline in accuracy. Since the memory required to store these maps grows quadratically with resolution, a resolution of 0.5 or 1.0 is recommended. Exploring the use of variable grid size is left for future work.

REFERENCES

[1] Y. Shi, W. Zhang, F. Li, and Q. Huang, "Robust localization system fusing vision and lidar under severe occlusion," *IEEE Access*, vol. 8, pp. 62 495–62 504, 2020.

[2] A. H. A. Hafez, M. Singh, K. M. Krishna, and C. V. Jawahar, "Visual localization in highly crowded urban environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2778–2783.

[3] R. Bao, R. Komatsu, R. Miyagusuku, M. Chino, A. Yamashita, and H. Asama, "Stereo camera visual slam with hierarchical masking and motion-state classification at outdoor construction sites containing large dynamic objects," *Advanced Robotics*, vol. 35, no. 3-4, pp. 228–241, 2021.

[4] B. Ferris, D. Hähnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proceeding of robotics: science and systems*, 2006.

[5] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A survey of machine learning for indoor positioning," *IEEE Access*, vol. 8, pp. 214 945–214 965, 2020.

[6] G. M. Mendoza-Silva, J. Torres-Sospedra, and J. Huerta, "A meta-review of indoor positioning systems," *Sensors*, vol. 19, no. 20, 2019.

[7] V. Gómez-Verdejo, E. Parrado-Hernández, and M. Martínez-Ramón, "Adaptive sparse gaussian process," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 11, pp. 16 383–16 395, 2023.

[8] M. Bauer, M. van der Wilk, and C. E. Rasmussen, "Understanding probabilistic sparse gaussian process approximations," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[9] T. Takebayashi, R. Miyagusuku, and K. Ozaki, "Development of magnetic-based navigation by constructing maps using machine learning for autonomous mobile robots in real environments," *Sensors*, vol. 21, no. 12, 2021.

[10] N. D. Lawrence and A. J. Moore, "Hierarchical gaussian process latent variable models," in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 481–488.

[11] H. Liu, J. Cai, Y. Wang, and Y. S. Ong, "Generalized robust bayesian committee machine for large-scale gaussian process regression," in *International Conference on Machine Learning*, 2018, pp. 3131–3140.

[12] P. Pfaff, R. Triebel, and W. Burgard, "An efficient extension to elevation maps for outdoor terrain mapping and loop closing," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 217–230, 2007.

[13] Y. Li, T. Jiang, X. Ding, and Y. Wang, "Location-free csi based activity recognition with angle difference of arrival," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[14] R. Miyagusuku, K. Tabata, and K. Ozaki, "Modeling robot orientation using wifi directional antennas and von mises distributions," in *2024 IEEE/SICE International Symposium on System Integration (SII)*, 2024, pp. 953–958.

[15] R. Miyagusuku, A. Yamashita, and H. Asama, "Precise and accurate wireless signal strength mappings using gaussian processes and path loss models," *Robotics and Autonomous Systems*, vol. 103, pp. 134–150, 2018.

[16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[17] S. Ito, F. Endres, M. Kuderer, G. D. Tipaldi, C. Stachniss, and W. Burgard, "W-rgb-d: Floor-plan-based indoor global localization using a depth camera and wifi," in *2014 IEEE international conference on robotics and automation (ICRA)*, 2014, pp. 417–422.

[18] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006.