

# Adaptive-Resolution Gaussian Process Mapping for Efficient UAV-based Terrain Monitoring

Liren Jin<sup>1</sup>, Julius Rückin<sup>1</sup>, Stefan H. Kiss<sup>2</sup>, Teresa Vidal-Calleja<sup>2</sup>, Marija Popović<sup>1</sup>

**Abstract**—Unmanned aerial vehicles (UAVs) are rapidly gaining popularity in a variety of environmental monitoring tasks. A key requirement for autonomous operation is the ability to perform efficient environmental mapping and path planning online, given their limited on-board resources constraining operation time and computational capacity. To address this, we present an adaptive-resolution approach for terrain mapping based on the Nd-tree structure and Gaussian Processes (GPs). Our approach enables retaining details in areas of interest using higher map resolutions while compressing information in uninteresting areas at coarser resolutions to achieve a compact map representation of the environment. A key aspect of our approach is an integral kernel encoding spatial correlation of 2D grid cells, which enables merging uninteresting grid cells in a theoretically sound way. Results show that our approach is more efficient in terms of time and memory consumption without compromising on mapping quality. The resulting adaptive-resolution map accelerates the on-line adaptive path planning as well. Both performance enhancement in mapping and planning facilitate the efficiency of autonomous environmental monitoring with UAVs.

## I. INTRODUCTION

Environmental monitoring is a core task in many application areas, including precision agriculture, post-disaster inspection, and industrial surveillance. In recent years, unmanned aerial vehicles (UAVs) have emerged as a more flexible, economic tool for environmental data acquisition compared to traditional methods, such as manual sampling or static sensor networks [1–3], which are limited in reliability and safety. With recent developments in on-board sensing and processing technologies, UAVs are experiencing a rapid uptake in a wide range of scenarios, e.g. biomass monitoring [2, 4], rescue missions [5, 6], temperature mapping [2, 7], and many more.

To fully exploit the advantages of UAV platforms, a key challenge is planning paths for efficient data-gathering given their physical constraints on operation time or battery life. Despite current advances in robot autonomy, most UAV missions are still performed in a precomputed, passive manner, e.g. using coverage path planning [8] to uniformly collect measurements about an environment. The major drawback of these approaches is that they cannot be easily modified to allow for targeted monitoring of objects or regions of interest, e.g. temperature hotspots, as they are discovered.

To overcome this issue, online informative path planning (IPP) strategies [5, 6, 9, 10] have recently been proposed to adaptively replan paths based on obtained measurements to maximize information gain from future measurements.

<sup>1</sup>Cluster of Excellence PhenoRob, Institute of Geodesy and Geoinformation, University of Bonn. <sup>2</sup>UTS Robotics Institute, Faculty of Engineering and IT, University of Technology Sydney. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2070 – 390732324. Corresponding: ljin@uni-bonn.de.

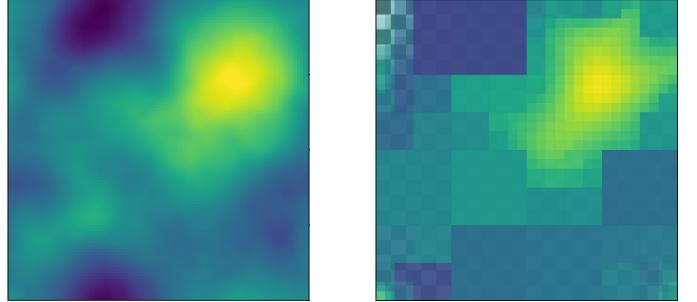


Fig 1: Our Adaptive-Resolution Gaussian Process strategy for online terrain mapping. Left: Ground truth field distribution on a terrain. Right: Our approach enables mapping areas of interest (yellow) with high resolution and low map uncertainty (low opacity) while compressing the map regions with low information value (blue). The checkerboard is added for visual interpretation of the opacity.

These methods rely on a map representation capturing the current environment state to guide next sensing actions. To support online map updates for fast decision-making, efficient computation and memory usage are thus a key prerequisite for mapping in autonomous monitoring tasks. Moreover, the mapping algorithms should not compromise on map quality, as inaccurate results can mislead next actions, e.g. waste valuable battery life on inspecting regions with low information value.

This paper focuses on online mapping methods in UAV-based terrain monitoring scenarios, where the goal is to recover a continuous 2D scalar field, e.g. of temperature, biomass cover, etc., using measurement information from on-board sensors. Different from our problem setup, most existing approaches in the remote sensing community exploit UAV-acquired data to construct very high-resolution terrain reconstructions, e.g. orthomosaic maps [2, 3]. Although such procedures yield highly detailed models, they often involve heavy post-processing and are thus not suitable for online mapping and adaptive planning. Our motivation is to develop an efficient, compact online mapping strategy to extend the applicability of UAVs for adaptive decision-making. Previous studies in IPP have tackled this problem by using fixed-resolution grid maps [10, 11] to represent terrain for replanning; however, they tend to scale poorly in large environments.

We propose a method for online terrain mapping called Adaptive-Resolution Gaussian Process (ARGP) mapping to address this problem. Our key insight is to adjust the map resolution based on the informativeness of associated measurements. Inspired by Popović et al. [4], we exploit a GP model to capture the spatial correlations in the underlying field, and use it as a prior for recursive data fusion with probabilistic sensors. Our ARGP mapping strategy extends this approach by introducing a new integral kernel and using an Nd-tree structure to adapt the map resolution online. This

enables us to preserve high-resolution details in interesting, i.e. highly informative, areas of the terrain, while using coarser resolutions otherwise as shown in Fig. 1. This way, we achieve memory- and computation-efficient mapping, which is necessary for autonomous decision-making on resource-constrained platforms. The contribution of this work is twofold:

- 1) A terrain mapping method based on Nd-tree data structure, which enables changing mapping resolution based on incoming data on-the-fly.
- 2) The introduction of a new integral kernel function to encode the spatial correlation of 2D grid cells in a terrain map. This enables us to merge grid cells at any scale to compress information in uninteresting regions.

We evaluate our mapping approach using synthetic and real-world data. Experimental results show that our method reduces memory consumption and improves computational efficiency in applications requiring online mapping and adaptive IPP.

## II. RELATED WORK

Our mapping approach is based on GPs and Nd-tree structure. This section reviews relevant work in these fields of study.

### A. Gaussian Processes Mapping

Occupancy grid maps are the most commonly used representation for robotic mapping [12]. Despite their successful application, traditional occupancy grid models assume the stochastic independence of grid cells to enhance computational efficiency [13]. However, this representation often poorly captures the spatial correlations found in natural physical phenomena, e.g. distributions of temperature, humidity, etc. on terrain. To address this, GPs models are often applied in many mapping applications. For instance, GPs are used to incorporate uncertainty and represent spatially correlated data to build a 2.5D pipe thickness map [14]. Vasudevan et al. [15] apply GP regression to estimate and interpolate elevation information on the field where sensory information is incomplete. Other applications include gas distribution mapping [16], occupancy mapping [13] and aquatic monitoring [17]. Our work follows these lines by using GPs to model a continuous scalar field.

The main limitation of applying GPs for online mapping is the cubic growing computational complexity with accumulating measurements over time. Note that the distant measurements have less influence on the grid cell value than nearby measurements. Therefore, GPs can be approximated through the use of local models. In [13, 15, 18], the measurement data are stored in a KD-tree structure. For calculating the predicted mean and variance of each grid cell, only nearby measurements are considered. However, local GP regression requires regression to be performed for individual test points one by one [19]. A different approach is to use recursive filtering to fuse new measurements incrementally into the map, so that it is unnecessary to preserve the measurement history and infer the posterior distribution from scratch each time new data arrive [20]. We use the same Bayesian update approach proposed in [14] to sequentially fuse new measurements into our map during a mission.

### B. Multi-Resolution Mapping

In many real-world terrain monitoring scenarios, the distribution of information in the environment is not uniform, i.e. some areas are considered more informative for mapping than others. Therefore, maintaining a map with constant resolution over the whole terrain is inefficient. A common method to generate more compact map representations is using tree structures. A well-known algorithm in this category is OctoMap, which is based on the octree structure [21]. Each node in the octree represents the space contained in a voxel. This voxel is then recursively divided into eight sub-volumes until the maximal resolution is reached. If all child nodes have the same state, e.g. occupied, the nodes are pruned, resulting in both memory-savings and the preservation of highly precise maps. Funk et al. [22] use the octree structure in an online mapping system that adjusts the map resolution based on occupancy state. Similarly, Chen et al. [23] apply quadtree to build multi-resolution 2D maps. Nd-tree generalizes these approaches by subdividing any d-dimensional volume recursively with  $N^d$  children [24, 25]. Rather than compressing the maps only after they have already been built, as in the case of OctoMap, we change the map resolution online based on incoming measurements in a similar way to Funk et al. [22], Einhorn et al. [25]. This enables us to tailor the map structure to conserve memory and computation time in applications requiring online mapping, such as adaptive path planning [9, 10].

## III. APPROACH

This section introduces our ARGP mapping approach. We define the map as a GP and store it in an Nd-tree. This uncertain prior GP initializes the recursive Bayesian fusion update from sequential measurements. We first present the theory behind GPs with the integral kernel function and an ‘average measurement’ model, in which the value of a grid cell represents the average value of a latent function in the area which it covers. This allows us to merge grid cells without losing important information during mapping. Then, we explain our Bayesian fusion update procedure and the merging operation for building multi-resolution terrain maps.

### A. Gaussian Processes and Integral Kernel

A GP is the generalization of a Gaussian distribution over a finite vector space to an infinite-dimensional function space. It can be fully described by its mean  $\mu(\mathbf{x})$  and covariance function  $k(\mathbf{x}, \mathbf{x}')$ . In practice, a GP is used to model spatial correlations in a probabilistic non-parametric manner and infer function values at a finite set of query points from the observed data [11]. The mapping target in our problem is assumed to be a continuous function:  $f(\mathbf{x}) \sim \mathcal{GP}(\mu, k) : \mathcal{E} \rightarrow \mathbb{R}$ , where  $\mathcal{E} \subset \mathbb{R}^2$  is the 2D rectangular input space and  $\mathbf{x} \in \mathcal{E}$ . We first sequentially discretize the input space into grid cells using Nd-tree until maximal depth  $t$  is reached. Note that only the leaf grid cells are shown in the grid cells map  $C = \{c_1, \dots, c_n\}$ , where  $n = (N^d)^t$  with  $d = 2$  as we focus on 2D terrain monitoring;  $c_i = [x_i^{\min}, x_i^{\max}] \times [y_i^{\min}, y_i^{\max}]$  is the parametrization of grid cell  $c_i \subset \mathcal{E}$ . We also define  $C = [c_1, \dots, c_n]^{\top}$  as the vectorization of  $C$ .

Given a pre-trained kernel function  $k(\mathbf{x}, \mathbf{x}')$ , we could obtain the correlation between the function value at any two points in our input space. However, we need to modify the learned kernel function to correctly encode the spatial correlations between grid cells in our mapping framework. Similarly, in [26], the author uses an integral kernel to incorporate line segment measurements into GP. Applying a linear operator to a GP leads to another GP [27]. We define  $\zeta(R) = \frac{1}{A} \int_R f(\mathbf{x}) d\mathbf{x}$  to represent the average of the latent function  $f$  over rectangular domain  $R \subset \mathbb{R}^2$  with area  $A \in \mathbb{R}$ . This way, we obtain the new GP  $\zeta(R) \sim \mathcal{GP}(\mu_1, k_1)$ , whose mean and covariance function are described as follows:

$$\mu_1(R_i) = \frac{1}{A_i} \int_{R_i} \mu(\mathbf{x}) d\mathbf{x}, \quad (1)$$

$$k_1(R_i, R_j) = \frac{1}{A_i A_j} \iint_{R_i \times R_j} k(\mathbf{x}, \mathbf{x}') d\mathbf{x} d\mathbf{x}', \quad (2)$$

where  $\mathbf{x}$  and  $\mathbf{x}'$  are the 2D positions contained within the rectangular domains  $R_i$  with area  $A_i$  and  $R_j$  with area  $A_j$  respectively. The area-related terms in Eqs. (1) and (2) simply transform the integration into average, which makes the physical meaning of our mean and covariance in accordance with our measurement model introduced in Sec. III-B.

### B. Sensor Model

A core characteristic of using UAVs for data acquisition is that on-board sensors can collect measurements from different altitudes. At each sensing location, the UAV obtains measurements from the downward-facing sensor, e.g. camera. The field of view (FoV) of the sensor is proportional to the altitude of the UAV, i.e. at higher altitude, the sensor provides larger FoV. Obviously, UAVs can conduct more time-efficient monitoring at higher altitudes, as less measurements and shorter travel distance are necessary to cover a fixed-size environment. However, since the measurements are noisy in real world, we should also consider the degraded performance of sensor at higher altitudes. We use Gaussian sensor model to encode this phenomenon into our Bayesian fusion update. For each observed grid cell  $c_i \in C$ , the sensor provides a measurement  $z_i$  capturing the average value of function  $f$  within this cell as  $z_i \sim \mathcal{N}(\mu_{s,i}, \sigma_{s,i}^2)$ , where  $\sigma_{s,i}^2$  is the noise variance expressing uncertainty in  $z_i$ . This noise variance can be decomposed into two parts described as follows.

First, we assume that the information collected by the sensor degrades with altitude. The measurements obtained from higher altitudes are more error-prone than those from closer detection. The sensor noise is described by  $\sigma_{a,i}^2 = \alpha h$ , where  $\alpha \in \mathbb{R}^+$  and  $h$  is the sensor's altitude.

Second, we also consider uncertainty caused by partially observing grid cells. In our mapping framework, the measurements obtained from a given grid cell are summarized as one single value representing the averaged function value. However, it is not rare that some grid cells are only partially covered by the FoV, especially when the grid cells occupy larger area after merging operation. Directly assigning the average measurements to these grid cells would be an over-confident assumption, as the unobserved part of these grid

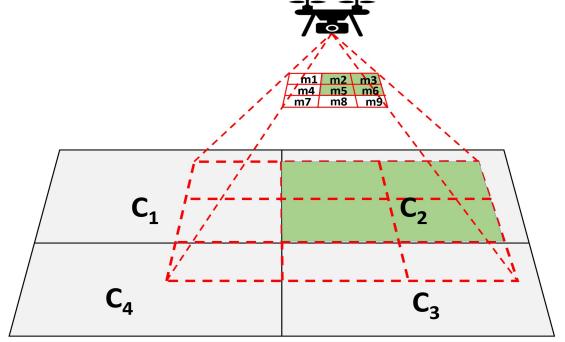


Fig 2: The ‘average’ sensor model provides the measurements of averaged function value over a grid cell. For instance, the measurement  $z_2$  observed from  $c_2$  is the average of 4 single measurement values  $\{m_2, m_3, m_5, m_6\}$ . For calculation of  $\sigma_{c,i}^2$ ,  $A_{\text{cover}}$  is the green area on the terrain and  $A_c$  is the area of  $c_2$  itself.

cells may contradict the current measurements, e.g. when grid cells contain both interesting and uninteresting regions. To tackle this problem, we propose the coverage-ratio-dependent variance  $\sigma_{c,i}^2 = \beta \left(1 - \frac{A_{\text{cover}}}{A_c}\right)$  in our sensor model, where  $\beta \in \mathbb{R}^+$  is a weighting and  $A_c, A_{\text{cover}}$  are the area of the grid cell and the part covered by the FoV. When a grid cell is fully observed in measurement  $z_i$ , this noise term is canceled out.

When the sensor takes a new measurement, the measurement data in our sensor model are generated as follows: given the known intrinsic and extrinsic parameters, and sensor height, the FoV can be determined. First, we query the grid cells which have overlap with the FoV using depth first tree search with pruning. Second, for observed grid cell  $c_i$ , we calculate the corresponding averaged measurement value  $z_i$  as illustrated in Fig. 2. Third, we sum the altitude-dependent variance  $\sigma_{a,i}^2$  and coverage-ratio-dependent variance  $\sigma_{c,i}^2$  as the total variance of each measurement  $z_i$ .

### C. Sequential Data Fusion

The map state is fully described by the mean vector  $\boldsymbol{\mu}(C)$  and covariance matrix  $\mathbf{K}(C, C)$ . The initial state can be obtained from Eqs. (1) and (2), which is then used as a prior for sequentially fusing new measurements:

$$\boldsymbol{\mu}^- = \begin{bmatrix} \mu_1(c_1) \\ \vdots \\ \mu_1(c_n) \end{bmatrix} \quad \mathbf{K}^- = \begin{bmatrix} k_1(c_1, c_1) & \dots & k_1(c_1, c_n) \\ \vdots & \ddots & \vdots \\ k_1(c_n, c_1) & \dots & k_1(c_n, c_n) \end{bmatrix} \quad (3)$$

We define  $\mathbf{z}$  as a vector consisting of  $m$  new average function value measurements observed from  $m$  corresponding grid cells as introduced above. To compute the posterior density  $p(\zeta | \mathbf{z}, C) \propto p(\mathbf{z} | \zeta, C) \cdot p(\zeta | C)$ , we directly apply the Kalman Filter update equations [20]:

$$\boldsymbol{\mu}^+ = \boldsymbol{\mu}^- + \boldsymbol{\Gamma} \mathbf{v}, \quad (4)$$

$$\mathbf{K}^+ = \mathbf{K}^- - \boldsymbol{\Gamma} \mathbf{H} \mathbf{K}^-, \quad (5)$$

where  $\boldsymbol{\Gamma} = \mathbf{K}^- \mathbf{H}^\top \mathbf{S}^{-1}$  is the Kalman gain, and  $\mathbf{v} = \mathbf{z} - \mathbf{H} \boldsymbol{\mu}^-$  and  $\mathbf{S} = \mathbf{H} \mathbf{K}^- \mathbf{H}^\top + \mathbf{R}$  are the measurement and covariance innovations.  $\mathbf{R}$  is diagonal  $m \times m$  matrix composed of variance term  $\sigma_{a,i}^2 + \sigma_{c,i}^2$  associated with each measurement  $z_i$  and  $\mathbf{H}$  is a  $m \times n$  matrix denoting the part of the map observed by  $\mathbf{z}$ . Note that at each measurement update, only  $\mathbf{S}$

is inverted, which is usually fast as  $m \ll n$ . However, when the covariance matrix is large, its multiplication becomes the main computational bottleneck of Kalman update. This motivates our map merging operation introduced in the following subsection.

#### D. Merging

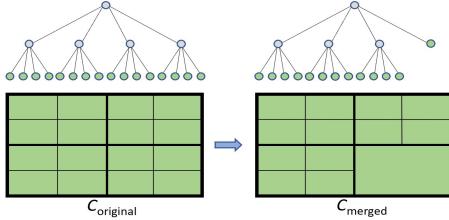


Fig 3: The Nd-tree (with  $N = d = 2$ ) structure and its corresponding grid cells map. Only leaf nodes (green) in the tree are shown in the map. After merging operation, the children grid cells are replaced by their parent in the new map.

A key requirement for multi-resolution mapping is the ability to manipulate the grid cell sizes on-the-fly. Our goal is to use coarser (larger) grid cells to map uninteresting regions and denser (smaller) grid cells to retain details of interesting parts. As the map is saved in a tree structure, we prune the children grid cells and merge the information into their parent. This merging operation allows us to summarize information in larger areas and thus reduce the total number of grid cells, which facilitates an efficient map update and memory usage for online path planning. For this, we subdivide our map into uninteresting regions (UR) and hotspots (HS):

$$C_{UR} = \{c_i \in C \mid \mu_i + \gamma K_{i,i} \leq f_{th}\}, C_{HS} = C \setminus C_{UR}, \quad (6)$$

where  $\mu_i$  and  $K_{i,i}$  are the mean and variance of grid cell  $c_i$ ; the design parameter  $\gamma$  is chosen to specify the margin to the threshold  $f_{th}$ . This setting avoids merging grid cells with possibly high mean values, which would cause detail loss in interesting regions. Thus, grid cells are only merged if there is a strong belief that they fall within uninteresting bounds.

For a parent grid cell, if all of its  $P = N^d$  child grid cells are uninteresting leaves (grid cells in  $C_{UI}$ ), these child grid cells can be pruned and replaced by their parent grid cell. A key challenge in our approach compared to other multi-resolution mapping algorithms using the independence assumption is that we need to adjust the covariance matrix during the merging operation. When we merge the information of  $P$  children into their parent grid cell, based on the definition of grid cell variable and the correlation encoded by the integral kernel, we can define the parent grid cell as:

$$\zeta_{\text{parent}} = \frac{1}{P} \sum_{i=1}^P \zeta_{\text{child}_i}. \quad (7)$$

The parent grid cell now represents the average function value of the whole region covered by its children. For the whole map, the merging operation can be described as a linear transformation of the GP, shown as follows:

$$\mu(C_{\text{merged}}) = M\mu(C_{\text{original}}), \quad (8)$$

$$K(C_{\text{merged}}, C_{\text{merged}}) = MK(C_{\text{original}}, C_{\text{original}})M^\top. \quad (9)$$

where  $C_{\text{original}}$  represents the vectorized map including  $n$  grid cells before merging and  $C_{\text{merged}}$  is the new vectorized map consisting of  $n - (P - 1) \times g$  grid cells, where  $g$  is the number of parent nodes whose children can be merged.

In the simplest case, where only one parent's children cells are merged,  $C_{\text{merged}}$ ,  $C_{\text{original}}$  and  $M$  can be expressed as:

$$C_{\text{original}} = \begin{bmatrix} c_1 \\ \vdots \\ c_{n-P} \\ c_{n-P+1} \\ \vdots \\ c_n \end{bmatrix}, \quad C_{\text{merged}} = \begin{bmatrix} c_1 \\ \vdots \\ c_{n-P} \\ c_{n-P+1} \end{bmatrix}, \quad (10)$$

$$M = \begin{bmatrix} I_{(n-P) \times (n-P)} & 0_{(n-P) \times P} \\ 0_{1 \times (n-P)} & Q_{1 \times P} \end{bmatrix}, \quad (11)$$

assuming that  $c_{n-P+1}$  in  $C_{\text{merged}}$  represents the parent of grid cells  $\{c_{n-P+1}, \dots, c_n\}$  in  $C_{\text{original}}$  and  $Q$  is  $\left[\frac{1}{P}, \dots, \frac{1}{P}\right]$ . A simple illustration is given in Fig. 3. The merging operation is performed after every measurement update for eligible grid cells. As the linear transformation of GP leads to another GP, we treat the resulting GP after the merging operation as a prior map for the next Bayesian update cycle.

## IV. EXPERIMENTAL RESULTS

In this section, our ARGP mapping method is evaluated in simulation by comparing it to different benchmarks. We then integrate our approach into an IPP framework to demonstrate its benefits for online mapping in adaptive terrain monitoring tasks using synthetic and real-world data.

### A. Mapping Performance

First, we evaluate our ARGP mapping approach against benchmarks in terms of computational and memory performance and mapping quality. To this end, we investigate four different mapping approaches:

- *Fixed-Resolution Mapping (FR)*: GP-based fixed-resolution mapping using Bayesian fusion update, as used by Popović et al. [4];
- *Gaussian Process Regression (GPR)*: fixed-resolution mapping using Gaussian Process regression directly [11];
- *Mapping under Independence Assumption (In.)*: fixed-resolution mapping assuming no correlations between grid cells;
- *ARGP*: our new GP-based multi-resolution mapping strategy using quadtree and Bayesian fusion update, as described in Sec. III.

We simulate 30  $20m \times 20m$  Gaussian random fields as ground truth environments. The ground truth maps are regularly discretized into  $200 \times 200$  grid cells using a uniform resolution of  $0.1m$ . For simplification, the ground truth field values are normalized to the interval  $[0, 1]$  and we define regions where ground truth values are above 0.7 as hotspots of interest. To study mapping performance at different map scales, we conduct the experiments at 3 different maximal resolutions:  $16 \times 16$ ,  $32 \times 32$  and  $64 \times 64$  grid cell maps corresponding to

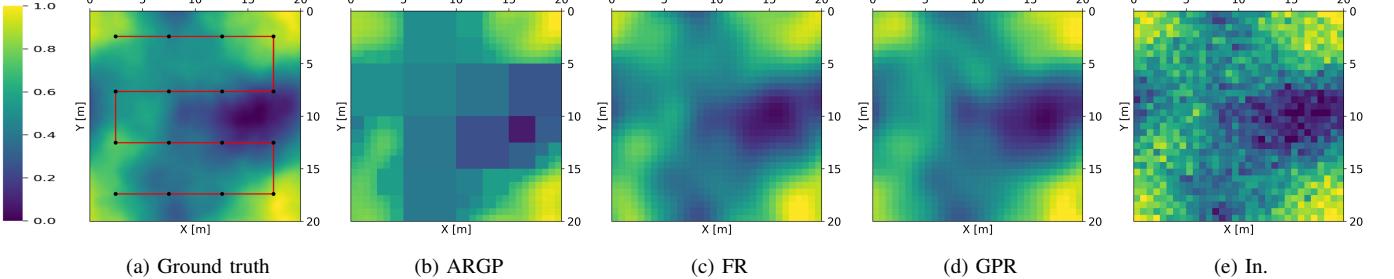


Fig 4: Qualitative comparison of our proposed ARGP approach (b) against benchmarks (c)-(e). The terrain is mapped using a lawnmower strategy, as shown in (a). The red line and black dots indicate the travelled path and measurement locations. All approaches use a map size of  $32 \times 32$  grid cells. By mapping adaptively, our method compresses information in areas with low information value (blue) while preserving details in higher-value areas of interest (yellow) to achieve an efficient, compact map representation for online applications.

Map size	Method	RMSE ↓	RMSE (hotspots) ↓	Mapping time [ms] ↓	Memory usage ratio ↓
16	ARGP	$0.089 \pm 0.011$	$0.065 \pm 0.008$	$35.8 \pm 7.4$	$0.231 \pm 0.13$
	FR	$0.063 \pm 0.004$	$0.062 \pm 0.007$	$35.3 \pm 7.7$	$1 \pm 0$
	GPR	$0.062 \pm 0.005$	$0.062 \pm 0.007$	$340.9 \pm 10.1$	$0.142 \pm 0$
	In.	$0.103 \pm 0.004$	$0.105 \pm 0.010$	$13.5 \pm 1.9$	$0.042 \pm 0$
32	ARGP	$0.083 \pm 0.011$	$0.038 \pm 0.006$	$321.4 \pm 51.2$	$0.055 \pm 0.08$
	FR	$0.036 \pm 0.003$	$0.036 \pm 0.004$	$685.5 \pm 31.1$	$1 \pm 0$
	GPR	$0.036 \pm 0.003$	$0.037 \pm 0.004$	$4732.1 \pm 22.5$	$0.036 \pm 0$
	In.	$0.098 \pm 0.002$	$0.097 \pm 0.004$	$37.2 \pm 5.2$	$0.011 \pm 0$
64	ARGP	$0.082 \pm 0.017$	$0.022 \pm 0.010$	$3074.4 \pm 300.6$	$0.007 \pm 0.003$
	FR	$0.022 \pm 0.001$	$0.022 \pm 0.002$	$9840.1 \pm 271.8$	$1 \pm 0$
	GPR	$0.022 \pm 0.001$	$0.023 \pm 0.002$	$1.2E05 \pm 193.7$	$0.009 \pm 0$
	In.	$0.097 \pm 0.001$	$0.097 \pm 0.002$	$73.7 \pm 4.5$	$0.003 \pm 0$

TABLE I: Comparison of our ARGP against benchmarks for varying map sizes. By adapting the map resolution, our strategy reduces runtime while delivering highly accurate maps. Note that memory usage is reported as a ratio relative to the FR approach.

maximal tree depths of 2, 3 and 4 respectively, in a quadtree configuration. Note that a quadtree is a special case of an Nd-tree, thus different mapping settings are possible by adapting the general Nd-tree decomposition.

The terrains are mapped using a lawnmower pattern to focus on comparing the methods in terms of mapping performance only, without variations due to planning. The UAV takes 16 non-overlapping measurements as shown in Fig. 4a to fully cover the terrain, assuming a flight altitude of  $2.5\text{m}$  and  $5\text{m} \times 5\text{m}$  FoV on the ground. The UAV starting position is  $[2.5, 2.5, 2.5]\text{m}$ . For the GP-based mapping approaches (*FR*, *GPR*, *ARGP*), we apply the squared exponential kernel function with hyperparameters  $\theta = \{\sigma^2, l\} = \{0.25, 2.36\}$  trained from the ground truth maps and following Eq. (2) to calculate the correlation in the prior maps. We choose a constant prior mean of 0.5. For mapping under independence assumption (*In.*), we generate the same prior map except setting all the cross-correlation terms to 0 to isolate each grid cell. The ‘average measurements’ model is utilized in all mapping approaches. For merging cells in *ARGP* mapping, we choose  $\{\gamma, f_{\text{th}}\} = \{2, 0.7\}$  in Eq. (6). We evaluate the mapping performance with respect to total mapping time, mapping accuracy in terms of root mean square error (RMSE) and memory consumption relative to *FR*. The total mapping time is calculated by aggregating the individual mapping times resulting from each measurement update. RMSE is obtained

by comparing the resulting maps with the whole ground truth map. The RMSE of hotspots only considers regions with ground truth values above 0.7.

The evaluation results are summarized in Tab. I and Fig. 4. In all cases, *In.* yields least accurate maps with highest RMSE since it is most sensitive to measurement noise due to missing correlation information. In contrast, the other three approaches reflect the smooth structure of the Gaussian random field, as they incorporate covariance information into the map update. As expected, the averaging effect caused by merging cells in *ARGP* leads to higher total RMSE compared to *FR* and *GPR*. However, *ARGP* shows the same accuracy in mapping hotspots, as required in our application scenarios. In terms of mapping efficiency, *GPR* performs worst as it considers all accumulated measurements to perform mapping operations, leading to prohibitively slow updates in online application. For small-scale maps, *ARGP* shows no improvement in time efficiency compared to *FR* due to the overhead caused by merging operation and tree search. However, as the major bottleneck in mapping is the large matrix multiplication in Eqs. (4) and (5). This overhead can be compensated by faster Bayesian fusion update with less grid cells in our approach. Another strength of our approach is that it is more memory-saving than *FR*; this is because it spares unnecessary memory consumption, e.g. large covariance matrix, by reducing the number of grid cells used to map uninteresting regions.

### B. Mapping for Path Planning

Next, we assess our approach in an adaptive planning context. We consider the same experimental setup as described in Sec. IV-A except setting our prior mean to 0.7 to initially encourage exploration in planning, and same ground truth map as shown in Fig. 1. We directly compare the *FR* and *ARGP* in the map scale of  $32 \times 32$ . For planning approach, we use greedy planning to find the next best measurement site. This is done by simulating map update in all possible measurements sites in the action space and choosing the one providing the largest reward, which is defined as:

$$r = \frac{\text{Tr}(\mathbf{K}^-) - \text{Tr}(\mathbf{K}^+)}{J(\mathbf{p}^-, \mathbf{p}^+)}, \quad (12)$$

where  $\mathbf{K}^-$ ,  $\mathbf{p}^-$  represent current covariance matrix and UAV position,  $\mathbf{K}^+$  is the covariance matrix after simulated mapping update in next measurement position  $\mathbf{p}^+$ . Note that both

covariance matrices are restricted to  $C_{HS}$  defined in Eq. (6). The denominator is the cost derived from flight time.

We use a 3D lattice above the terrain to represent the discrete action space. The lattice consists of measurement points at two different altitudes:  $\{2, 8\}m$ . At each altitude, we place the points such that the aggregation of the FoV fully covers the mapping region. The initial UAV position is set to  $[0, 20, 8]m$ . The total effective mission time should not exceed a given time budget of 100s. We use a constant velocity ( $v = 2m/s$ ) model to calculate flight times. The effective mission time is formulated as:  $t_{\text{mission}} = \sum_{i=1}^L t_{\text{mapping},i} + t_{\text{planning},i} + t_{\text{flight},i}$ , where  $L$  is the number of total measurement updates;  $t_{\text{mapping}}$ ,  $t_{\text{planning}}$ ,  $t_{\text{flight}}$  represent the time consumption on mapping update, greedy path planning and flight time to next measurement position respectively.

The experiment results are given in Fig. 5. The greedy planning using our ARGP mapping approach shows more efficient trace reduction behavior. This benefit becomes more pronounced in large action spaces and longer missions. Faster mapping update shortens the time consumption on forward prediction during planning. With shorter planning time, the UAV can achieve more efficient exploration and save time budget for exploiting observed hotspots, thus obtain lower uncertainty in these regions.

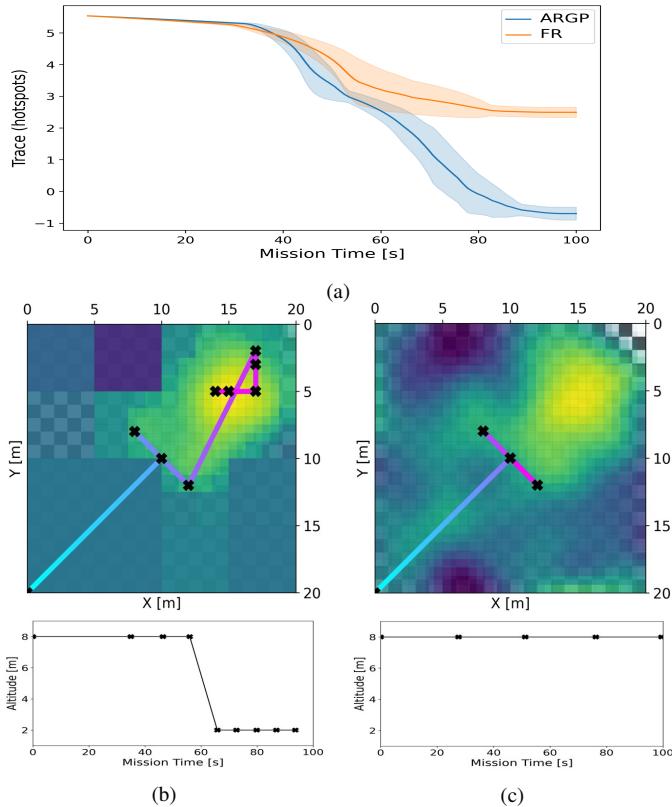


Fig 5: Comparison of our adaptive-resolution mapping approach (ARGP) against the fixed-resolution (FR) benchmark. (a) shows the mean and standard deviation of map uncertainty (logarithmic scale) in hotspots over 10 trials; (b) and (c) represent the UAV paths and mapping results using ARGP and FR respectively. The color of paths indicate time elapsed during the mission (from blue to red). High opacity indicates low variance in the maps. The bottom plots show the UAV altitude at each measurement site.

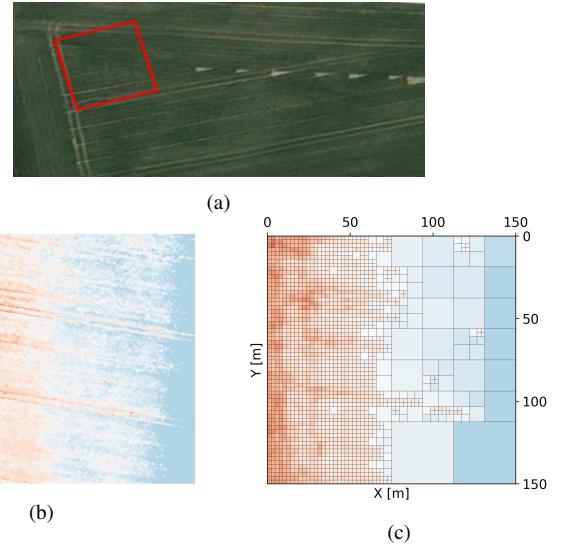


Fig 6: Validation of our ARGP approach for surface temperature mapping. The red box in (a) shows the experiment field where the temperature data (b) was collected. (c) shows the result of mapping this area using our method. High-temperature areas of interest (red) are mapped at higher resolutions to preserve information.

### C. Experiment with Real-World Data

We demonstrate our ARGP mapping approach using real-world surface temperature data. The data was collected in a  $150m \times 150m$  crop field ( $50.86^\circ$  lat.,  $6.44^\circ$  lon.) near Jülich, Germany on July 20, 2021 using a DJI Matrice 600 UAV platform equipped with a Vue Pro R 640 thermal sensor. During data acquisition, the UAV followed a lawnmower path at  $100m$  altitude to collect images at  $15cm$  spatial ground resolution. The images were processed using Pix4D software to create an orthomosaic used as ground truth in our mapping experiment. The aim is to validate our method for adaptively mapping temperature hotspots (higher than  $28^\circ\text{C}$ ) at finer resolutions using this real data. The mapping result in Fig. 6 confirms that our approach is capable of adapting grid cell resolution in a targeted way.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose ARGP mapping approach for online UAV-based terrain monitoring. We introduce a novel use of an integral GP kernel for capturing the correlation of two rectangular grid cells and use Nd-tree to store our map. This enables us to adapt the map resolution on-the-fly. Our results show that our approach is more efficient than traditional fixed-resolution mapping in terms of mapping time and memory consumption. Faster mapping update also facilitates adaptive path planning, thus yields a more efficient performance in autonomous terrain monitoring tasks using UAVs. The performance of our approach was validated using real-world data in a surface temperature mapping scenario. Future work will study more scalable and multi-robot mapping applications.

## ACKNOWLEDGEMENT

We would like to thank Mr. Jordan Bates from Forschungszentrum Jülich for providing real-world data and sensor information.

## REFERENCES

- [1] M. Dunbabin and L. Marques, "Robots for Environmental Monitoring: Significant Advancements and Applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.
- [2] S. Manfreda, M. F. McCabe, P. E. Miller, R. Lucas, V. Pajuelo Madrigal, G. Mallinis, E. Ben Dor, D. Helman, L. Estes, G. Ciralo et al., "On the Use of Unmanned Aerial Systems for Environmental Monitoring," *Remote Sensing*, vol. 10, no. 4, 2018.
- [3] G. Tmušić, S. Manfreda, H. Aasen, M. R. James, G. Gonçalves, E. Ben-Dor, A. Brook, M. Polinova, J. J. Arranz, J. Mészáros, R. Zhuang, K. Johansen, Y. Malbeteau, I. P. de Lima, C. Davids, S. Herban, and M. F. McCabe, "Current Practices in UAS-based Environmental Monitoring," *Remote Sensing*, vol. 12, no. 6, 2020.
- [4] M. Popović, T. Vidal-Calleja, G. Hitz, I. Sa, R. Siegwart, and J. Nieto, "Multiresolution mapping and informative path planning for UAV-based terrain monitoring," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 1382–1388.
- [5] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active Autonomous Aerial Exploration for Ground Robot Path Planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.
- [6] Z. W. Lim, D. Hsu, and W. S. Lee, "Adaptive informative path planning in metric spaces," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 585–598, 2016.
- [7] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, vol. 40, pp. 1059–1078, 2016.
- [8] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [9] M. Popović, T. Vidal-Calleja, G. Hitz, J. J. Chung, I. Sa, R. Siegwart, and J. Nieto, "An informative path planning framework for UAV-based terrain monitoring," *Autonomous Robots*, vol. 44, pp. 889–911, 2020.
- [10] G. Hitz, E. Galceran, M. É. Garneau, F. Pomerleau, and R. Siegwart, "Adaptive continuous-space informative path planning for online environmental monitoring," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1427–1449, 2017.
- [11] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [12] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [13] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, 2012.
- [14] T. Vidal-Calleja, D. Su, F. De Bruijn, and J. V. Miro, "Learning spatial correlations for Bayesian fusion in pipe thickness mapping," in *IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 683–690.
- [15] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair, "Gaussian Process modeling of large scale terrain," in *IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 1047–1053.
- [16] C. Stachniss, C. Plagemann, and A. Lilienthal, "Learning Gas Distribution Models using Sparse Gaussian Process Mixtures," *Autonomous Robots*, vol. 26, pp. 187–202, 2009.
- [17] G. Hitz, A. Gotovos, F. Pomerleau, M.-E. Garneau, C. Pradalier, A. Krause, and R. Siegwart, "Fully Autonomous Focused Exploration for Robotic Environmental Monitoring," in *IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 2658–2664.
- [18] Y. Shen, A. Ng, and M. Seeger, "Fast Gaussian Process Regression using KD-Trees," *Neural Information Processing Systems*, vol. 18, 2005.
- [19] J. Wang and B. Englot, "Fast, accurate Gaussian process occupancy maps via test-data octrees and nested Bayesian fusion," in *IEEE International Conference on Robotics and Automation*. IEEE, 2016, pp. 1003–1010.
- [20] S. Reece and S. Roberts, "An introduction to Gaussian processes for the Kalman filter expert," in *International Conference on Information Fusion*. IEEE, 2010.
- [21] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [22] N. Funk, J. Tarrio, S. Papatheodorou, M. Popović, P. F. Alcantarilla, and S. Leutenegger, "Multi-Resolution 3D Mapping with Explicit Free Space Representation for Fast and Accurate Mobile Robot Motion Planning," *Robotics and Automation Letters*, vol. 6, pp. 3553–3560, 2021.
- [23] Y. Chen, W. Shuai, and X. Chen, "A probabilistic, variable-resolution and effective quadtree representation for mapping of large environments," in *International Conference on Advanced Robotics*. IEEE, 2015, pp. 605–610.
- [24] E. Einhorn and C. Schr, "Building 2D and 3D adaptive-resolution occupancy maps using ND-Trees," *Robotics*, pp. 306–311, 2010.
- [25] E. Einhorn, C. Schröter, and H. M. Gross, "Finding the adequate resolution for grid mapping - Cell sizes locally adapting on-the-fly," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1843–1848.
- [26] S. O'Callaghan and F. Ramos, "Continuous Occupancy Mapping with Integral Kernels," in *AAAI Conference on Artificial Intelligence*. AAAI, 2011.
- [27] S. Särkkä, "Linear operators and stochastic partial differential equations in Gaussian process regression," *Lecture Notes in Computer Science*, vol. 6792, no. 2, pp. 151–158, 2011.