sensors

MDPI

Article

An Integrated Strategy for Autonomous Exploration of Spatial Processes in Unknown Environments

Valentina Karolj [1,*], Alberto Viseras [2], Luis Merino [1] and Dmitriy Shutin [2]

[1] Service Robotics Laboratory, Universidad Pablo de Olavide, Crta. Utrera km 1, 41013 Seville, Spain; lmercab@upo.es

[2] German Aerospace Center, Oberpfaffenhofen, 82234 Weßling, Germany; alberto.viserasruiz@dlr.de (A.V.); dmitriy.shutin@dlr.de (D.S.)

* Correspondence: vkar@alu.upo.es

Received: 3 June 2020; Accepted: 27 June 2020; Published: 30 June 2020

Abstract: Exploration of spatial processes, such as radioactivity or temperature is a fundamental task in many robotic applications. In the literature, robotic exploration is mainly carried out for applications where the environment is a priori known. However, for most real life applications this assumption often does not hold, specifically for disaster scenarios. In this paper, we propose a novel integrated strategy that allows a robot to explore a spatial process of interest in an unknown environment. To this end, we build upon two major blocks. First, we propose the use of GP to model the spatial process of interest, and process entropy to drive the exploration. Second, we employ registration algorithms for robot mapping and localization, and frontier-based exploration to explore the environment. However, map and process exploration can be conflicting goals. Our integrated strategy fuses the two aforementioned blocks through a trade-off between process and map exploration. We carry out extensive evaluations of our algorithm in simulated environments with respect to different baselines and environment setups using simulated GP data as a process at hand. Additionally, we perform experimental verification with a mobile holonomic robot exploring a simulated process in an unknown labyrinth environment. Demonstrated results show that our integrated strategy outperforms both frontier-based and GP entropy-driven exploration strategies.

Keywords: Gaussian process; exploration; mapping; autonomous robots; mobile robots; information gathering

1. Introduction

1.1. Motivation

Autonomous robots are well suited for gathering information in hostile or hardly accessible environments. For instance, they can be used to map radiation or gas concentration levels in NBC hazards [1,2], temperature and/or humidity fields, etc. If the environment is partially or fully unknown, a robot needs to explore and build a 2D/3D obstacle map to ensure that the whole environment is covered. At the same time, it needs to safely navigate through the operating environment and localize itself and the gathered information with respect to such a map in the absence of global localization.

Over the last few decades, mapping and SLAM have been areas of extensive research in the robotics community for obtaining accurate maps of unknown environments [3]. Moreover, the extension to active mapping and active SLAM has been considered [4,5]. While active mapping determines the actions of the robot that improve area coverage [6], active SLAM considers at the same time maximization of coverage while maintaining localization performance [7].

In many cases, active mapping and active SLAM are required for proper localization of the robot, construction of an accurate map, and navigation purposes. These are essential support for the main mission of the robot, e.g., monitoring radiation, magnetic field, temperature distribution, etc. The main problem considered here is how a robot can efficiently carry out this main mission (explore such fields) while building at the same time the map of the environment.

Ideas similar to active mapping can be used to explore the spatial field [8]. However, this can lead to potentially conflicting goals, given that different models and sensors (with a different FOV, etc.) are used for building the 2D/3D obstacle map and for field reconstruction. This paper thus deals with developing techniques for autonomous exploration of such processes in completely unknown environments, which consider at the same time and in an integrated way efficient map building and process estimation. We focus on exploring physical processes that can be modelled as GP [9]. Examples of such a process are temperature, ozone concentration, and magnetic field intensity, to name only a few.

We challenge here two fundamental assumptions that are typically considered in the state of the art: (i) A priori knowledge about the operating environment (the map), and (ii) that process and map exploration can be solved independently. The aim of our integrated strategy is to minimize the error in the process reconstruction and to maximize the coverage of the environment map, while minimizing the total travelled distance of the exploratory mission all together.

1.2. Related Work

Recent advances in mobile robotics have opened new frontiers for the development of novel exploration algorithms. In the literature, approaches for robot exploration are typically based on the maximum informativeness criterion (e.g., [8,10–12]), which guides the robot towards locations with the highest information gain.

We are interested in complex probabilistic models that allow us to model environmental state and learn the process from acquired data while exploiting the spatial correlation between variables of the process in order to obtain faster exploration results. GPs represent a powerful approach to model spatial phenomena. Singh et al. [11] propose a procedure to define suitable covariance functions for Gaussian process regression in environmental surveillance applications. Then they extend those covariance functions to perform informative sampling often combined with GPs due to their ability to predict the remaining uncertainty about the process.

Krause et al. study what is the optimal placement for a network of sensors in order to reduce the uncertainty of the GP [10]. They consider the sensor placements as fixed. In contrast, we are interested in exploration with a mobile robot. This allows us to cover larger environments, gives us more flexibility to monitor dynamic processes, and re-plan in newly discovered areas.

Informative sampling that takes into account an information metric (i.e., mutual information that takes into account statistical dependencies between sampling point candidates) in combination with GPs has been considered for different exploration applications, e.g., magnetic field mapping [8], environmental monitoring [13], and online estimation of a radio signal source [14]. However, the works in [8,13,14] assume the environment in which the robot operates as a priori known. In order to deploy autonomous robots in complex environments, we must develop algorithms that are able to operate in unknown environments.

Active SLAM approaches tackle the operation in unknown environments while considering joint uncertainty reduction for map and robot pose trajectory. Common frameworks consider information theoretical approaches where decision is based on the notion of information gain, guiding the robot towards locations where information about unknown environment can be obtained [5,15]. Occupancy grids are commonly used for map representation and navigation purposes [16]. Besides occupancy grids, GPs are introduced as continuous parametrization of frontiers for autonomous exploration in combination with SLAM [17]. There, the authors use BCM to incrementally build GP maps of the environment while guiding the robot toward borders with unknown regions on the map. However, the aforementioned approaches do not incorporate the exploration of a spatial process within the framework of a geometrical map exploration.

Approaches, where SLAM and a spatially distributed process are jointly considered, are exemplified in [18]. There, the authors describe a gas distribution mapping algorithm which is able to consider and propagate the uncertainty in the pose estimate, integrating it with SLAM to create two independent occupancy grids for (i) environment and (ii) gas distribution. However, the exploration strategy is not considered, and the system is teleoperated.

Besides occupancy maps, other spatial processes could be used as anchors for geometric SLAM. For instance, Jung et al. [19] introduce magnetic field constraints for pose-SLAM optimization for loop-closure and heading correction with off-line processing of the gathered magnetic sensor data for creation of the pose-SLAM. As in the previous case, active mapping component is not considered.

Prágr et al. [20] propose a greedy approach that combines spatial exploration (mapping) with the exploration of the underlying traversal cost model over terrain by employing the RBCM. Their approach is similar to ours with respect to the combination of GP and frontier-based exploration. However, a key difference is that we consider a GP that is not necessarily correlated with the spatial map—by mapping new regions of the environment via perception sensor, we do not gain additional information about the process like in their case. With GP reconstruction, useful knowledge of the process increases only in the vicinity where measurements are obtained. A second key difference is that, in contrast to the aforementioned greedy strategy [20], we present a multi-step horizon planning for the integrated exploration that takes into account spatial field exploration on a smaller scale than the environment mapping.

An interesting approach to the multi-step-ahead exploration strategy is the use of TSP to model the goal selection problem. Travelled distance can be described as a cost to be minimized while visiting all goals. Kulich et al. [21] introduce a TSP-based approach for finding a minimum distance path to all frontiers instead of greedily visiting frontiers one by one, benefitting in a possible shorter exploration time. Oßwald et al. [22] apply a similar strategy by exploiting prior knowledge and running a TSP on an user-defined topological graph. They combine this solution with local exploration of individual locations and the results demonstrate a significant reduction in exploration time. Authors in [23] apply constrained generalized TSP to solve map coverage while integrating goal selection with the goal candidates generation. In this work, we apply a similar method for minimizing distance in several steps of our exploration strategy.

In contrast to map coverage in [21–23], our focus is on the joint exploration of map and GP. The exploration objectives include a multi-step exploration strategy, considering that the map has to be fully covered, but at the same time the process exploration task has to be fulfilled. This can be viewed as a specific case of a multi-objective optimization for the most informative candidate points selection that are (i) extending map coverage, (ii) decreasing process error.

There has been extensive work on multi-objective approaches that combine exploration in unknown environments and mapping [24–26]. Classical approaches towards a solution to this complex problem consider various perspectives e.g., (i) aggregation into a single utility function [15,24,27]; (ii) Pareto efficiency, i.e., improving one objective will negatively affect the others [25]; (iii) domain specific robot exploration and search problem represented as a high-level state machine—Petri nets [26]. Aggregation of different criteria (i.e., distance, information gain, localization uncertainty, communication, etc.) has mostly ad hoc design due to sensitivity to combined criteria, and it requires careful weight prioritisation. Stachniss and Burgard [27] apply a weight tuning to combine the uncertainty reduction with the distance travelled. Carrillo et al. [15] introduce an automatically tuned factor for a trade-off in exploration–exploitation dilemma in SLAM. Despite overcoming weight set-up, their approach cannot be generalized to process exploration—it is task specific for combining localization and mapping. A more general multi-criteria approach is proposed as aggregation operator obtained from Choquet integral [28], and weighted means by Basilico and Amigoni [24]. They apply multi-criteria decision making theoretical approach for searching fixed targets (victims) in a priori unknown environment. Such an approach offers more flexibility for composing multi-objective criteria; however it still has the drawback of weight setting. Amigoni and Gallo [25] solve this problem of weighting utilities as a multi-objective optimization by finding a Pareto front as the best trade-off between competing objectives for a multi-robot system. Candidates are Pareto efficient with respect to travelling cost, information gain, and expected overlap between robots. The best position is greedily chosen from the Pareto frontier as the nearest distance from the ideal solution (i.e., best travelling cost, best information gain, and best overlap). Their approach to find a good simultaneous solution for all objectives is interesting, however it does not guarantee that the chosen candidate is the best among Pareto efficient candidates. A more intuitive, high level representation of multi-objective optimization for exploration and search strategies is presented by Calisi et al. [26]. They represent robot plans as Petri nets, adapted to the tasks of exploration, search and rescue missions. Petri nets offer a modular approach in resolving complex situations such as loops, interrupts, etc. Authors apply it for map exploration while assuming that interesting features can be always intercepted by the same sensor used for mapping. With that assumption, it is only needed to search for victims along the obstacles detected by the laser scanner while avoiding unnecessary search in open space. To the contrary with Calisi et al. [26], we need to explicitly go into open space since the obstacle perception sensor does not provide us knowledge about the process, which is collected with a point-wise sensor.

To the best of our knowledge, there is a research gap in combining mapping and process exploration strategies. Particularly in cases where, given sensors on a different scale, obtained data from perception sensor does not produce any new information about the underlying spatial process we want to explore.

1.3. Contribution

The main contribution of this paper is a novel integrated strategy for map and physical process exploration in an unknown environment with the following characteristics:

- It considers multi-step exploration, which has proven to be superior in comparison with greedy, one-step strategies [21,22].
- It selects intermediate goals in multi-step exploration for efficiently exploring the environment, while reducing the reconstruction error of the spatial process.
- It imposes visitation of intermediate goals as a routing problem for minimizing the traversed distance between two multi-exploration steps.
- It combines the strategy with efficient modelling using the GRBCM [29] to maintain online computational capabilities when exploring larger areas.

The remainder of the paper is organized as follows. Representation of a spatial process as GP is discussed in Section 2. A detailed description of the introduced system and its functionalities are given in Section 3, along with the exploration objectives. Evaluation setup and results followed by discussion are presented in Sections 4–6. Conclusion of the paper is given in Section 7, where the proposed method and its possibilities for further development are discussed.

2. Gaussian Process Regression

2.1. Gaussian Process

A GP is any finite collection of random variables that has a joint multivariate Gaussian distribution [9]. It is a very popular and well-known non-parametric statistical method to model spatial dependencies of a physical process. It is a distribution over functions [30] fully characterized by its mean function $\mathbf{m}(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}', \boldsymbol{\theta})$ for any pair $\mathbf{x}$, $\mathbf{x}'$ representing spatial locations and some hyperparameters $\boldsymbol{\theta}$ [9].

We assume that the value of $\mathbf{m}(\mathbf{x})$ is set to zero, which implies a priori no known values of the observed process. In this work, we are interested in modelling smooth processes. Accordingly, we employ the covariance function as a SE [9]:

$$k(\mathbf{x},\mathbf{x}',\boldsymbol{\theta}) = \sigma_{\mathrm{f}}^2 \exp(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2l^2}) + \sigma_{\mathrm{n}}^2 \delta_{\mathbf{x}\mathbf{x}'}, \tag{1}$$

where $\delta_{\mathbf{x}\mathbf{x}'}$ is Kronecker's delta, and $\boldsymbol{\theta} = [\sigma_{\mathrm{f}}^2, l, \sigma_{\mathrm{n}}^2]^T$ represents hyperparameters. For different covariance functions (Matérn, Rational Quadratic, etc.) and their characteristics, we refer the reader to [9].

SE hyperparameters are defined as follows: (i) The signal standard deviation σ_{f}^2 dictates the variation of function values from the mean; (ii) the characteristic length-scale l represents how smooth the function is and informally "how close" two points $\mathbf{x}$ and $\mathbf{x}'$ have to be for influencing each other significantly; (iii) variance of the sensor noise fluctuations σ_{n}^2 represents expected level of noise present in the data [9].

Next, we present the following definitions: $\mathbf{X} = [\mathbf{x}^{[1]}, \mathbf{x}^{[2]}, \cdots, \mathbf{x}^{[N]}]^T$ is a matrix where each row corresponds to a spatial location where measurement data was acquired by the robot, $\mathbf{z} = [z^{[1]}, z^{[2]}, \cdots, z^{[N]}]^T$ are the corresponding N measurements, and $\mathbf{X}_* = [\mathbf{x}_*^{[1]}, \mathbf{x}_*^{[2]}, \cdots, \mathbf{x}_*^{[R]}]^T$ is a matrix where rows correspond to R "test" locations—spatial points where we want to predict process values using the learned model.

Now we use $k(\mathbf{x},\mathbf{x}',\boldsymbol{\theta})$ to define matrices that evaluate cross-correlations: (i) $\mathbf{K}$—between measurement points $\mathbf{X}$; (ii) $\mathbf{K}_*$ —between measurements $\mathbf{X}$ and "test" points $\mathbf{X}_*$; (iii) $\mathbf{K}_{**}$ —between "test" points $\mathbf{X}_*$. As a remark, matrices $\mathbf{K}$, $\mathbf{K}_*$, and $\mathbf{K}_{**}$ are all implicit functions of $\boldsymbol{\theta}$ due to their dependency on the covariance function $k(\mathbf{x},\mathbf{x}',\boldsymbol{\theta})$.

Our aim is to predict process values $\mathbf{y}_*$ and the corresponding uncertainties at "test" locations $\mathbf{X}_*$, given obtained measurements $\mathbf{z}$ and associated locations $\mathbf{X}$. Before being able to predict the process with the GP model, we need to learn it. This implies utilizing training data, $\mathbf{X}$ and $\mathbf{z}$, to determine the hyperparameters. Given the training data, we can estimate the suitable hyperparameters $\boldsymbol{\theta}_*$ for our measurements. The learning phase is also known as model training. The hyperparameter estimation is typically executed via gradient-based optimization aimed at maximizing the log-marginal likelihood function of $\boldsymbol{\theta}$ [9]:

$$\boldsymbol{\theta}_* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ -\frac{1}{2}\mathbf{z}^T\mathbf{K}^{-1}\mathbf{z} - \frac{1}{2}\log|\mathbf{K}| \right\}. \tag{2}$$

Finally, the prediction $\mathbf{y}_*$ is based on $\boldsymbol{\theta}_*$ (Equation (2)). The vector $\mathbf{y}_*$ is a random vector with the following probability density function: $p(\mathbf{y}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$, where $\boldsymbol{\mu}_*$, $\boldsymbol{\Sigma}_*$ are calculated as (more details in [9]):

$$\begin{aligned} \boldsymbol{\mu}_* &= \mathbf{m}(\mathbf{X}_*) + \mathbf{K}_*^T\mathbf{K}^{-1}(\mathbf{z} - \mathbf{m}(\mathbf{X})), \\ \boldsymbol{\Sigma}_* &= \mathbf{K}_{**} - \mathbf{K}_*^T\mathbf{K}^{-1}\mathbf{K}_*. \end{aligned} \tag{3}$$

The main drawback of using a GP in real-world applications is its poor scalability with respect to data size. Given N, its complexity during the training phase is $\mathcal{O}(N^3)$ as we need to invert the $N \times N$ dimensional covariance matrix. Moreover, due to matrix-vector operation, complexity of its prediction phase is $\mathcal{O}(N^2)$.

2.2. Large-Scale Gaussian Process Regression

In order to apply a GP without limiting the size of the operated environment, we apply GRBCM [29], an aggregation method that allows us to combine independently trained GP regressors from multiple data sets.

Firstly, given an initial dataset $\mathcal{D}$ of size N, it is divided into P subsets, each with size p_0: One communication disjoint partition $\mathcal{D}_{\mathrm{c}} = \mathcal{D}_1$ generated as a random subset, and disjoint partitions $\mathcal{D}_p = \{\mathbf{X}^{(p)}, \mathbf{z}^{(p)}\}, p \ \geq 2 \ldots P$ generated by k-means clustering [31]. Secondly, the data-sets $\{\mathcal{D}_p\}_p^P$, $p \geq 2$ in GRBCM are extended with a communication dataset as enhanced sets $\mathcal{D}_{+p} = \{\mathcal{D}_{\mathrm{c}}, \mathcal{D}_p\}$.

The GP regression (Equation (3)) is then applied on communication partition $\mathcal{D}_{\mathrm{c}}$, producing a global communication expert $\mathcal{E}_{\mathrm{c}}$ trained on $\mathcal{D}_{\mathrm{c}}$ that captures the main features of the target function. Furthermore, it is applied to each disjoint partition $\mathcal{D}_{+p}$, leading to the second expert type—local experts. Those experts $\{\mathcal{E}_p\}_{p=2}^{P}$ locally refine predictions trained on respective $\mathcal{D}_{+p}$.

It is important to mention the assumption that local experts $\mathcal{E}_p$ are uncorrelated, leading to the following predictive distribution equations for aggregated model [29]:

$$\begin{aligned} \mu_{*,\mathrm{A}}(\mathbf{x}_*) &= \sigma_{*,\mathrm{A}}(\mathbf{x}_*)\{\sum_{p=2}^{P} \beta_p \sigma_{+p}^{-2}(\mathbf{x}_*)\mu_{+p}(\mathbf{x}_*) - (\sum_{p=2}^{P} \beta_p - 1)\sigma_{\mathrm{c}}^{-2}(\mathbf{x}_*)\mu_{\mathrm{c}}(\mathbf{x}_*)\}, \\ \sigma_{*,\mathrm{A}}^{-1}(\mathbf{x}_*) &= \sum_{p=2}^{P} \sigma_{+p}^{-2}(\mathbf{x}_*) - (\sum_{p=2}^{P} \beta_p - 1)\sigma_{\mathrm{c}}^{-2}(\mathbf{x}_*), \end{aligned} \tag{4}$$

where β_p is defined as [29]:

$$\beta_p = \begin{cases} 1, & p = 2 \\ 0.5(\log \sigma_c^2(\mathbf{x}_*) - \log \sigma_{+p}^2(\mathbf{x}_*)), & 3 \leq p \leq P. \end{cases} \tag{5}$$

In summary, the global communication expert captures the main features of the model, while the local experts are improving local predictions. Generally, for aggregation models, calculating local experts principally has the same complexity as calculating a single global expert—cubic in model training and quadratic in prediction. However, a significant scalability improvement lies behind two reasons: (i) The size of local experts is much smaller than the full dataset N ($p_0 \ll N$), and (ii) local experts can be computed in parallel and in a distributed manner. For more details about GRBCM algorithm refer to [29].

3. Integrated Exploration

In this section we describe our proposed algorithm for autonomous exploration of a spatial process in an unknown environment. First, we present an overview of the system. Second, we describe the elements that compose our strategy in detail.

We aim to explore a spatial physical process $\mathbf{y}$ at R corresponding locations $\mathbf{X}_* \subset \mathbb{R}^{2\times R}$ where we want to obtain process values which can be modelled as a GP (Section 2.1). For large-scale spatial physical processes we construct the GP as an aggregated modelling of individual experts with GRBCM (Section 2.2).

Additionally, we assume that the structure of the operating environment is a priori unknown. That is, information about the position of obstacles is unavailable. The lack of information about the environment motivates us to extend process exploration with mapping. Note that an accurate map $\mathcal{M}$ is crucial for exploration planning and safe navigation through the environment. Furthermore, a robot must localize itself with respect to such a map by means of on-board sensors as an external localization system is unavailable. We would like to remark here that we assume the ability of the on-board localization system to maintain the position uncertainty below the process correlation length. This assumption permits us to focus on our aforementioned strategy. Meaning, that for considered environments, enhanced odometry is sufficient for localization. Nevertheless, we will study localization component as a part of our future work.

We consider a ground-based mobile robotic platform equipped with an on-board computer capable of real-time computations. It carries a 3D LIDAR that provides point clouds in the local sensor frame, and a magnetic field sensor that collects one process measurement $z^{[n]}$ (observed value of the process $\mathbf{y}$) per location $\mathbf{x}^{[n]}, n = 1, \ldots, N$.

Figure 1 shows an overview of the complete system. Inputs from on-board sensors are directly forwarded to the module for environment mapping (Mapping and Localization) and indirectly to the process estimation module (GP Estimator) through the exploration module (Integrated Exploration). Details of these two modules are explained in Section 3.2. Their output is used by the key module of our system—the Integrated Exploration module.

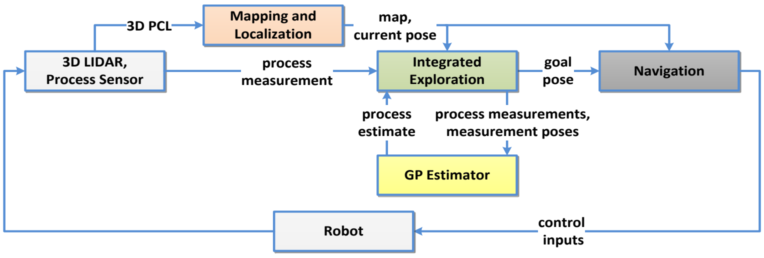


Figure 1. System overview.

The Integrated Exploration module has the following tasks: (i) Point of Interest (POI generation $\mathcal{X}_{\text{poi}} = \{\mathbf{x}^{[j]}, j \ = 1, ..., G\}$, where G is the number of POIs; and (ii) sort $\mathcal{X}_{\text{poi}}$ in a specific way to obtain an ordered sequence of POIs as goal poses $\mathbf{X}_{\text{goal}} = \{\mathbf{x}^{[g]}, g \ = 1, ..., G\}$. More details about the exploration module are provided in Section 3.4. Further, goal poses are forwarded sequentially to the Navigation module responsible for local path planning and execution by providing control inputs for autonomous movement of the robotic platform.

The robot starts its exploration of the environment at a certain known initial position $\mathbf{x}^{[0]}$, obtaining initial scan cloud $\mathcal{P}^{[0]}$ and process measurement $z^{[0]}$. During exploration, it continuously observes the environment through its perception sensor and localizes itself within the area. In parallel, the robot takes measurements of the process $z^{[g]}$ at locations $\mathbf{x}^{[g]} \in \mathbf{X}_{\text{goal}}$. This loop continues until a stopping criterion is fulfilled. For details of the algorithm description refer to Section 3.4.

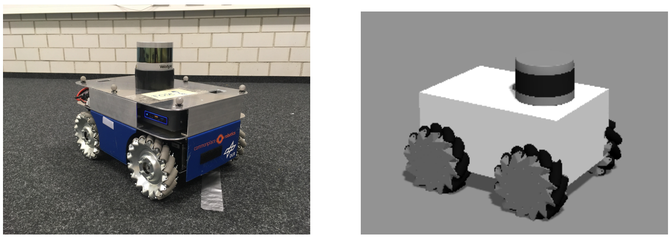
3.1. Sensors and Robot

3.1.1. Sensors

We employ two sensors: (i) Perception sensor—responsible for mapping the environment and obstacle avoidance, and (ii) process sensor—responsible for collecting spatial process information (i.e., magnetic field measurements). For the former, we have a 3D LIDAR sensor with 360° horizontal, and 30° vertical FOV with a range up to 100 m. However, any other perception sensor that provides distance from obstacles may be applied. For the latter, we consider a magnetic field sensor that collects information point-wise, i.e., only at the current location.

3.1.2. Robot

We use a holonomic robotic platform (Figure 2) equipped with sensors and an on-board computer for autonomous localization, mapping, navigation and exploration execution (Sections 3.2–3.4). Generally, any robotic system that explores in 2D can be applied if equipped with aforementioned required components.



(**a**) Slider (**b**) Simulated robot

Figure 2. Robotic platform equipped with Velodyne VLP-16 and an on-board computer.

3.2. Mapping and Localization, Navigation

3.2.1. Mapping

For navigation and exploration purposes, we employ an occupancy grid representation of the map $\mathcal{M}$. An occupancy grid map is a two-dimensional grid representation of the environment, holding a probabilistic information of each cell occupancy, which can be marked as: (i) Free, (ii) occupied, or (iii) no information [32].

To construct an occupancy grid, we use point cloud inputs $\mathcal{P}^{[1:t]}$, where t is current input time from the 3D LIDAR, which we then vertically project onto 2D xy horizontal plane given robot's height dimension constraints. Note that before projecting input clouds $\mathcal{P}^{[1:t]}$, we apply ground base filtering (by extraction of ground points) and height filtering (by extracting points above certain height). This projection is used for safe navigation through the environment with obstacles of various shape and size.

3.2.2. Localization

Pose $\mathbf{x}^{[t]}$ at time t is represented as a 2D position $[x, y]$ in Cartesian coordinate system, and yaw as orientation. Odometry and translation/rotation estimation of the incoming point cloud $\mathcal{P}^{[t]}$ is carried out with respect to the reference cloud $\mathcal{P}^{[t-s,t-1]}$ (cloud of processed, i.e., filtered and concatenated, s previously registered scans). For this, a registration algorithm—the point-to-plane variant of ICP algorithm [33]—is employed. Note that during exploration, relative localization with respect to initial pose $\mathbf{x}_0$ is performed. One can choose from a selection of readily available state-of-the-art vision and LIDAR-based localization/SLAM algorithms such as LOAM [34], Lego-LOAM [35], SegMap [36], or Cartographer [37] for maintaining localization accuracy in GPS-denied environments.

3.2.3. Navigation

We employ a combination of optimal global A* planner [38] on the grid with TEB reactive local planner [39]. Waypoints are fed to the Navigation module by the Integrated Exploration module (Section 3.4). Additionally, it receives periodical updates from Mapping and Localization module (previously described), assuring no-collision paths.

3.3. GP Estimator

At any time t, from the set of collected measurements $\mathbf{z}^{[1:t]} = [z^{[1]}, \ldots, z^{[t]}]^T$ and its estimated positions $\mathbf{X}^{[1:t]} = [\mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[t]}]^T$, we can predict the process values $\mathbf{y}_*$ and corresponding uncertainties at probe positions $\mathbf{X}_*$ (Equation (3)). Furthermore, in our case, hyperparameters are known and learned a priori. Let us remark that hyperparameters could be also learned online from measurements (Equation (2)).

As presented in Section 2.2, in order to cope with scalability, we extend our system with GRBCM [29]—a state-of-the-art aggregative method for the GP that synthesizes predictions from individually trained experts, where experts are trained on sub-sets of the full data set. Predictive distribution for GRBCM is estimated according to Equation (4).

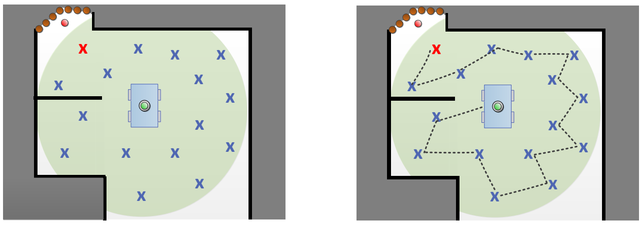
3.4. Exploration Strategy

The aim of our integrated strategy is to address the problem of reconstructing a spatial GP in an a priori unknown environment while simultaneously exploring it. We assess the behaviour of the integrated strategy with respect to the two main objectives and its performance towards minimizing the total travelled distance while satisfying both:

1. Efficient spatial process exploration: Minimization of the process error in comparison to the ground truth.

2. Efficient coverage strategy: Increasing coverage of the environment map to reduce unknown portions of the map.

As already mentioned, this integrated strategy consists of two key steps regarding POIs: (i) Identification (Section 3.4.1) and (ii) ordering—determining goals (Section 3.4.2). Results of both steps are visualized in Figure 3.



(**a**) Identified POIs (**b**) Goal poses

Figure 3. Main components of our exploration strategy. (**a**) Current position $\mathbf{x}^{[t]}$ is marked with a green circle, light red cross marks goal location $\mathbf{x}_G$ within planning radius r (shaded green area), blue crosses represent POI $\mathcal{X}_{\text{poi}}$, while orange circles are frontier with its centroid $\mathbf{x}_f$ (red circle); (**b**) optimized by distance travelled, $\mathcal{X}_{\text{poi}}$ are represented as ordered points, i.e., goal poses $\mathbf{X}_{\text{goal}}$, which describe the sequence that the robot follows, obtaining measurement $z^{[g]}$ on each $\mathbf{x}^{[g]}$.

3.4.1. POI Detection

In order to tackle our joined exploration objective, we select POIs in which taking measurements will improve estimate of the (i) current map (by considering frontiers of the map) and (ii) process (by selecting informative candidates on the estimated spatial field):

Frontier. First, to efficiently map the environment, we select classical frontiers, which maximize map coverage [6]. For efficient detection of frontiers, which separate known regions from unknown in $\mathcal{M}$, we apply WFD, a graph-based approach based on BFS [40]. This algorithm performs a search only on cells that are not yet traversed and represent free space, thus avoiding an expensive full map search. We weigh and sort frontiers by distance from the current location of the robot $\mathbf{x}^{[t]}$ to their centroids—the average position of all frontier points for a given frontier—and choosing the closest one. The distance is calculated with the A* path-planning algorithm.

The shortest path to the centroid $\mathbf{x}_f$ of the selected frontier provides a fast way to increase map coverage, but it drastically limits the area from which measurements of the process are obtainable. Since we can only collect process measurements point-wise given the sensor footprint, following the shortest way towards the frontier will not be the most informative path regarding process exploration. A lot of areas along the way will consequently be missed. Therefore, a trade-off between the shortest and the most informative path is considered by expanding the path from $\mathbf{x}^{[t]}$ towards $\mathbf{x}_f$ with a set of possible positions where the robot may obtain new informative process measurements while advancing the map coverage.

Candidates (POI). Searching for all possible process measurement locations $\mathcal{X}$ on $\mathcal{M}$ is infeasible for large maps. Moreover, we need to keep in mind that map and process estimations are changing with robot traversing forward, meaning that planning over a too long horizon may produce an inefficient exploration strategy due to significant changes in the map.

To overcome this, we limit the exploration of the process to a finite horizon r with respect to the robot position $\mathbf{x}^{[t]}$ at any given time t. To identify potential POI candidates with a high information value, we apply BFS for traversing only through obstacle-free explored cells in $\mathcal{M}$ within the horizon r. We sample $\mathcal{X}_{\text{poi}} \subseteq \mathcal{X}_{\text{free},r} \subseteq \mathcal{X}_r \subseteq \mathcal{X}$, where $\mathcal{X}_r$ represents all measurement locations within the radius, while $\mathcal{X}_{\text{free},r}$ represents its subset of reachable locations (Algorithm 1, below). Here, $\mathcal{X}_{\text{poi}}$ is the set of selected measurement locations where all the surrounding cells within the inflation radius are unoccupied. If the frontier centroid $\mathbf{x}_f$ is within radius r, it will be included in $\mathcal{X}_{\text{poi}}$.

As a metric for informativeness of each candidate, we define the information value of a position $\mathbf{x}_*$ as the Shannon entropy of the posterior of the GP at that position:

$$H(\mathbf{x}_*) = \frac{1}{2}\ln(2\pi e \sigma^2_{\mathbf{x}_*}), \tag{6}$$

where $\sigma^2_{\mathbf{x}_*}$ represents the variance of the estimated process $\mathbf{y}_*$ at location $\mathbf{x}_*$. A high value of entropy here represents a high uncertainty of the process (i.e., non-observed cell). Variance $\sigma^2_{\mathbf{x}_*}$ is estimated with GP regression (Section 2.1). Here, we emphasize that due to previously mentioned scalabilty limitations of classical GP regression approach, for a larger number of measurements we apply GRBCM that allows us to keep real-time performance of the algorithm.

Algorithm 1 $\mathcal{X}_{\text{poi}}$ informative candidates sampling.

```
Require: current location $\mathbf{x}^{[t]}$, map $\mathcal{M}$, radius $r$, sampling distance $k$, threshold $\epsilon$
Ensure: POI $\mathbf{X}_{\text{poi}}$
1: Extract $\mathcal{X}_{\text{free}}$ from the $\mathcal{M}$.
2: $\mathcal{X}_{\text{poi}} = \{\mathbf{x}^{[t]}\}$
3: BFS ($\mathcal{X}_{\text{free}}, \mathbf{x}^{[t]}$):
4:     $\hat{\mathbf{x}} = \{\mathbf{x}_* \in \mathcal{X}_{\text{free}} : H(\mathbf{x}_*) > \epsilon, \left\|\mathbf{x}_* - \mathbf{x}^{[p]}\right\| \geq k, \forall \mathbf{x}^{[p]} \in \mathcal{X}_{\text{poi}}, \}$
5:     $\mathcal{X}_{\text{poi}} = \mathcal{X}_{\text{poi}} \cup \{\hat{\mathbf{x}}\}$
```

We sample with a minimum separation distance k with respect to points $\mathbf{x}^{[p]}$ already in $\mathcal{X}_{\text{poi}}$, where k is chosen depending on the hyperparameters $\boldsymbol{\theta}$, particularly by l, a length-scale that defines a process correlation, informally representing "how far we can reliably extrapolate from obtained data" (Section 2.1). Values below threshold ϵ (also dependant on hyperparametrs) represent low entropy, i.e., low process uncertainty.

3.4.2. Goals Detection

Finally, once the POIs are selected, the robot needs to visit them all while minimizing the distance. To achieve this, POIs first need to be ordered as further described in this section. Ordered POIs are represented as goals $\mathbf{X}_{\text{goal}}$.

Distance matrix. In order to find the optimal solution to traversing all $\mathcal{X}_{\text{poi}}$ in terms of the distance travelled, we calculate the distance matrix between each pair of POIs $\mathbf{x}^{[i]}, \mathbf{x}^{[j]} \in \mathcal{X}_{\text{poi}}$, including the current pose $\mathbf{x}^{[t]}$. The distance between $\mathbf{x}^{[i]}$ and $\mathbf{x}^{[j]}$ is calculated with A* algorithm, where $i, j \in [1, \ldots, G]$.

Route calculation. Once the matrix is created, we formulate our problem as a routing problem. Since TSP does not consider distinguishable depots for both start and finish, we apply a simplified case of VRP [41]—a generalized version of TSP problem. More specifically, we apply a variant of the multi-depot VRP introducing additional constraints on start/end depots to minimize travelling distance from current location $\mathbf{x}^{[t]}$ to goal location $\mathbf{x}_G$ while visiting all $\mathcal{X}_{\text{poi}}$. We call it "vanilla" VRP with only one vehicle with given constraints to minimize the distance travelled between start and finishing depots.

The reason for imposing additional constraints on the depots (start and finish location) for the solver is to indicate that we are not interesting in returning to the starting location, but rather to a specific location aligned with our exploration objectives. Depots are determined as follows:

- As a start depot, set the current location $\mathbf{x}^{[t]}$.
- If the frontier centroid $\mathbf{x}_f \in \mathcal{X}_{\text{poi}}$, then $\mathbf{x}_G = \mathbf{x}_f$.

- Otherwise, we set $\mathbf{x}_G$ to be $\mathbf{x}^{[p]} \in \mathcal{X}_{\text{poi}}$, that has the shortest distance to frontier centroid $\mathbf{x}_f$ (to preserve the direction favoring area coverage).

The output of "vanilla" VRP solver is a vector $\mathbf{X}_{\text{goal}}$ of sorted waypoints. Figure 4 depicts a simple example of the solver solution where constraints are only applied on the exact starting and ending depot for a one-robot system.

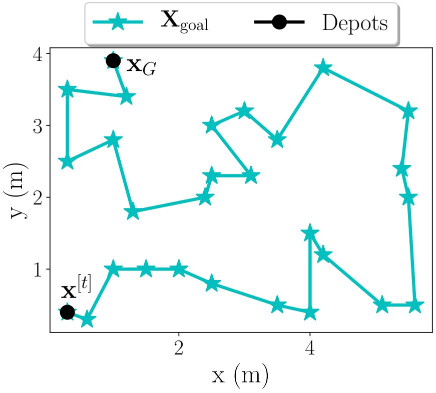


Figure 4. One-robot solving a multi-depot "vanilla" VRP: $\mathbf{x}^{[t]}$—current location; $\mathbf{x}_{\text{G}}$—goal location. Goals ($\mathbf{X}_{\text{goal}}$) are represented with blue asterisks and depots with black circles.

3.5. All Components of Our Integrated Exploration Strategy

Let us consider an example scenario where the robot has already partially explored the environment (Figure 5). The green circle represents the robot during the experiment, with current time t. The grey circles, along with the green one, represent visited locations $\mathbf{X}^{[1:t]} = [\mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[t]}]$ with their observed process value measurements $\mathbf{z}^{[1:t]} = [z^{[1]}, \ldots, z^{[t]}]$. Now, we also have an environment map $\mathcal{M}$ with regions marked as free, occupied or unknown. Given our planning horizon (radius r), all $\mathcal{X}_{\text{poi}}$ are detected (Section 3.4.1). This corresponds to the exploration objective of obtaining measurements at the most informative points. However, we still need to answer the question: "What is the most efficient order to visit them with objectives of minimizing the travelled distance and increasing the map coverage?" This is where the knowledge about frontiers comes in place. We can exploit the frontier in order to guide robot towards the goal position that is closest to the "border between the known and unknown" which guarantees an increase in map coverage (Section 3.4.2).

However, we need to deviate from the shortest path in order to collect process samples. Therefore, the start depot $\mathbf{x}^{[t]}$ and end depot $\mathbf{x}_{\text{G}}$ (guiding towards $\mathbf{x}_{\text{f}}$) are used as inputs to "vanilla" VRP. Note that if we would neglect knowledge about $\mathbf{x}_{\text{f}}$, the robot could finish its exploration e.g., in the far left bottom corner (Figure 5), which means it would need to traverse a significant distance towards unknown regions in order to proceed with exploration.

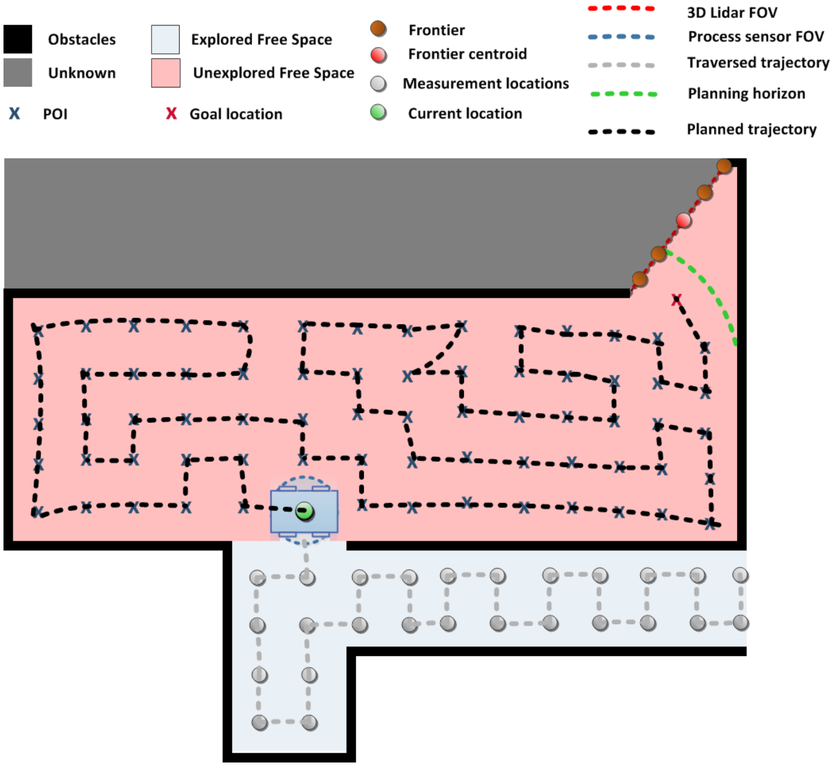


Figure 5. All components of a single multi-step of our integrated exploration strategy: Grey circles represent previously obtained measurements and the area around those circles is considered explored (light blue). Robot, at the current location (green circle), samples Point of Interest (POI, blue crosses)—in the obstacle-free region of the map, where the process is unexplored (red area)—within radius r (green line). The routing algorithm provides a multi-step exploration solution (black dotted line) from the current location to the goal location (red cross). The goal location is the POI closest to the frontier (brown circles) and the corresponding centroid (red circle). The frontier is defined by the LIDAR FOV (red dotted line). Obstacles are marked by a black line and unexplored area in grey.

Algorithm Work-Flow

Figure 6 describes the algorithm that controls the behaviour of the robot. We can summarize it as follows:

- Mapping and Localization: The robot continuously perceives the environment and accordingly updates the map $\mathcal{M}$ and its current location estimate $\mathbf{x}^{[t]}$.
- Navigation: Until any unvisited $\mathbf{x}^{[g]} \in \mathbf{X}_{\text{goal}}$ exists, it continues following precomputed goal poses $\mathbf{X}_{\text{goal}}$ (ordered representation of $\mathcal{X}_{\text{poi}}$).
- At each $\mathbf{x}^{[g]}$ reached, collect the process measurement $z^{[g]}$.
- GP Estimator: Estimate GP process at probe locations $\mathbf{X}_*$ over the whole environment.
- Integrated Exploration If $\mathbf{X}_{\text{goal}}$ is empty, detect the next frontier on $\mathcal{M}$ according to the procedure described in Section 3.4.1 and:
  - Sample locations within r as described in Section 3.4.1—producing unordered $\mathcal{X}_{\text{poi}}$, a list of candidates where we want to obtain our next measurements to increase knowledge about the process.
  - From $\mathcal{X}_{\text{poi}}$, create a distance matrix, representing computed distances between POI.
  - Order POI according to the procedure described in Section 3.4.2 so that all POI are visited and total travelled distance is minimized, resulting in $\mathbf{X}_{\text{goal}}$.

- If Algorithm 1 finds no suitable candidates within the limited horizon r, extend the horizon to cover all discovered cells $\mathcal{X}_{free}$ on the map. Select only the closest candidate location that satisfies $H(\mathbf{x}) > \epsilon$ as the next goal location $\mathbf{x}^{[g]}$. Otherwise terminate the mission.

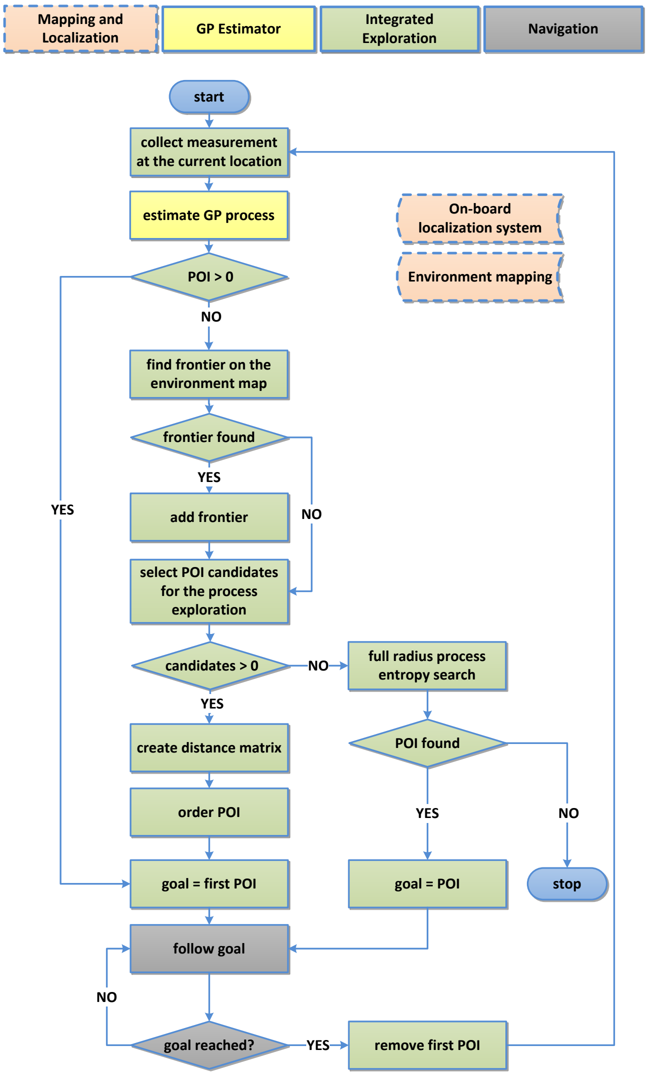


Figure 6. Algorithm state diagram. Colored blocks represent the following: Red—mapping and localization; yellow—estimation of GP; green—core algorithm (our integrated exploration); and grey—navigation.

4. System Evaluation

We first evaluate the proposed strategy in various simulation scenarios. Then we carry out an experiment, in which a mobile holonomic robot operates in an unknown environment populated by obstacles to explore magnetic field intensity. As a remark, during evaluation our proposed strategy will be referred to as Integrated Exploration (IE).

We assess the following:

1. What is the scalability of GRBCM for exploration of spatial processes?—Simulations (Section 5.1).
2. What is the correlation between sampling distance k and error decrease in the process reconstruction for the IE strategy? How does it affect total exploration distance?—Simulations, experiment (Sections 5.2 and 6.2).

3. How does the IE perform against the benchmarks in various scenarios?—Simulations, experiment (Sections 5.3.3 and 6.2).

The scenarios considered in the evaluation are listed in Table 1.

Table 1. Evaluated scenarios in simulation (Sx) and Experimental (Ex) test cases.

Scenario	Process	Operating Environment	Dimensions (m^2)
S1	Process 1	Obstacle-free	8.0 × 8.0
S2	Process 1	Small room-like environment, obstacles introduced	8.0 × 8.0
S3	Process 2	Large room-like environment, obstacles introduced	56.1 × 38.4
E1	Magnetic field intensity	Obstacles introduced	12.0 × 6.0

4.1. General System Setup

4.1.1. Robotic Platform

For experiments we use a holonomic ground robotic platform equipped with an Intel Nuc (Intel Nuc https://www.intel.de/) (Figure 2a), further referred to as a "slider". A simulated representation of the robot with mechanum wheels is made in Gazebo (Gazebo http://gazebosim.org/), based on the characteristics of the real-robotic platform (Figure 2b).

4.1.2. Perception Sensor

Mapping and localization is carried out with the 3D LIDAR Velodyne VLP-16 (Velodyne https://velodynelidar.com/vlp-16.html).

4.1.3. Process Sensor

Resolution of the simulated process sensor is 0.1 m and measurements $z^{[g]}$ per location $\mathbf{x}^{[g]}$ are obtained for various sets of predefined ground truth data which represent the GP (Table 2). The ground truth positions are estimated by Gazebo and the motion tracking system VICON (VICON https://www.vicon.com/) for simulations and experiments, respectively. Moreover, regarding Table 2, Process 1 and Process 2 measurements are samplings from a simulated multi-variant GP and the magnetic field data corresponds to a previously collected real-world magnetic field intensity [8].

Table 2. Explored GP process $\mathbf{y}$ with hyper-parameters $\boldsymbol{\theta}$.

Process	Dimensions (m^2)	σ_f^2(m)	l (m)	σ_n^2 (m)
Process 1	8.0 × 8.0	0.03	0.2	0.0001
Process 2	56.1 × 38.4	0.04	0.25	0.0001
Magnetic field	4.3 × 9.6	0.07	0.2	0.0001

Software components of the system are implemented using ROS (ROS—Robot Operating System. http://wiki.ros.org/.), Gazebo, C++, and Python 2.7. The process estimator is implemented by employing an open-source GP framework GPy [42]. Lastly, in order to solve the routing problem, we utilize the OR-Tools [43] (library developed and maintained by Google).

5. Simulations

5.1. Scalability of Gaussian Processes for Spatial Modelling

The aim of our first simulation tests is to validate the performance of GRBCM in predicting spatial processes in comparison with a full GP prediction.

In order to evaluate the GRBCM performance on a large data set, we consider a simulated 2D multivariate GP dataset with $N = 7592$ measurement points and $R = 214500$ (390 × 550) "test" locations. As a reminder, for the GRBCM we have $P - 1$ local experts $\mathcal{E}_p$ and one communication global expert $\mathcal{E}_c$. We assume the same size for all experts, i.e., for a dataset size N, the size of each is $p_0 = N/P$.

Simulation tests are performed on Dell XPS 13 7390 laptop with Linux operating system Ubuntu 18.04LTS, 10th Generation Intel(R) Core(TM) i7-10510U, and 16 GB RAM.

Simulation Results

We present results in Table 3 that show a significant improvement in the time required to estimate the process, with only a slight increase in the final NMSE of the reconstructed process. Moreover, it is observed that the NMSE error decreases when assigning more observations to each expert, i.e., increasing p_o. However, we note that more observations per expert lead to increased computation costs due to matrix inversion. Furthermore, we see that even with a smaller p_o, time does not necessarily decrease. This is due to the increased total number of experts P and multiple parallel computing processes. Therefore, by finding a good trade-off between the minimum size of experts and computational capabilities, we can obtain an accurate estimate for large-scale processes with prominently low computation and time costs.

In our further simulations, process estimation with GRBCM is carried out with a fixed expert size p_0 of 350. For $N < 350$, we have only one expert, which gives an equivalent performance to classical GP.

Table 3. Performance evaluation for 2D field estimation with dataset with $N = 7592$ measurements, R = 214,500 2D test points (390 × 550), grid resolution 0.1 m, hyperparameter $l = 0.25$ m, sampling $k = 3l$. We variate P and $p_0 = N/P$ (number of local experts and their size, respectively).

P, p_o	Method	Prediction Time [s]	*NMSE*
37, 100	GRBCM	21.28	0.151
18, 200	GRBCM	20.74	0.093
12, 300	GRBCM	20.26	0.073
10, 350	GRBCM	19.26	0.070
9, 400	GRBCM	22.42	0.068
7, 500	GRBCM	24.31	0.062
1, 7592	Full GP	102.34	0.060

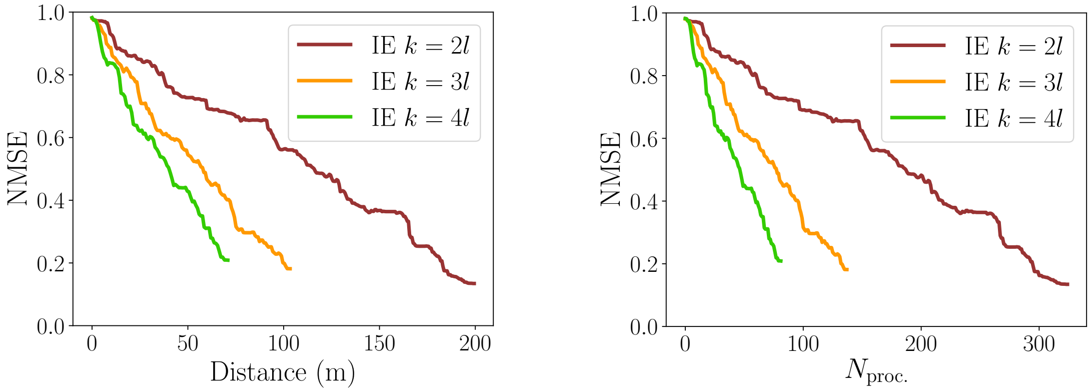
5.2. Sampling Distance

Since our aim is to consider real-world scenarios, we only want to collect important information. We ask ourselves what is the minimal distance between samples k that will lead to an accurate reconstruction of the process but at the same time balance the number of samples we need to obtain. We vary k with respect to the length-scale l. In such a scenario, we obtain observations about Process 1 (S1: Table 1) with $l = 0.2$ m, applying our IE strategy in an empty room with a full horizon. For all runs, the starting position is $(x(\text{m}), y\ (\text{m})) = (-3.3, -3.3)$.

Simulation Results

Figure 7 presents the results with a varying sampling step $k = [2l, 3l, 4l]$. With IE, the number of samples $N_{\text{proc.}} = [325, 138, 82]$ are obtained and the corresponding NMSE in relation to the ground truth are [0.13, 0.18, 0.21]. The travelled distances are [199.45 m, 103.57 m, 70.82 m]. We can notice that the higher the sampling distance k is, the steeper the NMSE curve is with respect to the distance (Figure 7a) and the number of samples N (Figure 7b). At the same time, the final attained NMSE is larger with larger distances. Let us remark that NMSE will never reach zero due to physical limitations of the robot, which is constrained not to collect data closer than 0.5 m to the obstacles in the environment.

For the rest of the experiments we select $k = 3l$, which gives us an optimal trade-off between final process accuracy and distance travelled.



(**a**) Process error vs. Distance

(**b**) Process error vs. Number of samples

Figure 7. Variation of sampling distance parameter k.

5.3. Evaluation of the Strategy in Simulation

5.3.1. Applied Baselines

In the literature for GP exploration, entropy-based methods [10,17,44] are typically used and they are showing dominance over random methods. Moreover, the baseline for map exploration originates from the frontier-based method [6], which is used in many map exploration (i.e., [17,20]) and active SLAM approaches (i.e., [5,15]).

Thus, in order to validate the performance of our integrated exploration strategy, we consider a combination of entropy driven and frontier-based baseline strategies as follows:

GGE. At each step, selecting one $\mathbf{x}_g$ from all possible reachable cells $\mathcal{X}_{free,r}$ within radius covering all discovered cells $r = full$:

$$\mathcal{X}_c = \left\{ \mathbf{x}_g : \mathbf{x}_g = \arg\max_{\mathbf{x} \in \mathcal{X}_c} H(\mathbf{x}),\ s.t. H(\mathbf{x}) > \epsilon \right\}, \tag{7}$$

with stopping criteria $\mathcal{X}_c = \{\}$, i.e., when there are no more cells satisfying Equation (7). This baseline is inspired by the maximum informativeness criterion [44] that guides the robot towards locations with the highest information gain while considering the global information available about the process.

GLGE. At each step, selecting one $\mathbf{x}_g$ from all possible reachable cells $\mathcal{X}_{free,r}$ within radius r covering all discovered cells that is the solution of:

$$\mathbf{x}_g = \arg\max_{\mathbf{x} \in \mathcal{X}_{free,r}} H(\mathbf{x}),\ s.t. H(\mathbf{x}) > \epsilon, \tag{8}$$

with stopping criteria $\mathbf{x}_g = \mathbf{0}$, i.e., when there are no more cells satisfying Equation (8). This baseline is likewise inspired by the maximum informativeness criterion [44] yet with a small twist that guides the robot towards locations with the information gain above a priori set threshold, while considering the global information available about the process only if it cannot find a local suitable candidate.

SS. Goal pose is $\mathbf{x}_g = \mathbf{x}_f$, the centroid of the frontier. If at least one reachable frontier exists, measurements are collected each k meters distance travelled towards $\mathbf{x}_g$, where parameter k is the sampling distance described in Section 3.4.1. Baseline for the map exploration frontier originates from commonly used frontier-based exploration, both in the map coverage tasks [6,17,20], as well as a part of active SLAM exploration component [5,15]. Once no more frontiers are presented, i.e., map is fully explored, the GLGE strategy is used to complete an entropy driven process exploration.

We would like to remark that the direct comparison with active SLAM algorithms that consider the localization uncertainty, regarding SS baseline, is not considered due to our assumption that the on-board localization system is able to localize the robot with satisfying accuracy, and therefore we neglect loop-closures and the need to revisit previously explored areas. A summary of the evaluated benchmarks along with IE strategy is presented in Table 4.

Table 4. Evaluated strategies compared against our Integrated Exploration (IE) strategy.

Strategy	Radius r (m)	Step Size k (m)	Multi-Step Planner
GGE	explored map	$[3l$, explored map$]$	No.
GLGE	(i) fixed r, (ii) explored map	(i) $[3l, r]$, (ii) $[3l$, explored map$]$	No.
SS	explored map	(i) fixed = $3l$, (ii) $[3l$, explored map$]$	No.
our IE	(i) fixed r, (ii) explored map	(i) $[3l, r]$, (ii) $[3l$, explored map$]$	Yes.

5.3.2. System Simulation Setup

To examine the behaviour of each benchmark against our strategy on a smaller scale, we consider a simulated physical process (Process 1: Table 2) with a priori given hyperparameters. Simulations are carried out in two 8 × 8 m^2 environments with distinct degrees of obstacle complexity: (i) Obstacle-free (S1: Table 1) and (ii) room-like (S2, S3: Table 1). For these environments, 5 starting positions are chosen: $(x(\mathrm{m}), y\ (\mathrm{m})) = (-3.3, 1.0)$, $(-3.3, -3.3)$, $(0.0, 0.0)$, $(2.0, -2.3)$, and $(2.5, -2.5)$.

Furthermore, to additionally evaluate our system on a larger scale, we test our IE strategy against GLGE in a large-scale environment (56.1 × 38.4 m^2) populated with obstacles, considering a simulated physical process (Process 2: Table 2) with a priori given hyperparameters.

The robot footprint, including a safety buffer, is set to radius of 0.5 m for all simulation scenarios.

Simulation tests S1, S2 (Table 1) are performed on computer with Linux operating system Ubuntu 16.04LTS, Intel(R) Core(TM) i7-3820, and 32GB RAM. Further simulation scenario S3 (Table 1) is performed on computer with Linux operating system Ubuntu 18.04LTS, Intel(R) Core(TM) i7-10510U, and 16GB RAM.

5.3.3. System Simulation Results

Obstacle-free small room-like environment (S1 in Table 1). We assume that the process is fully explored when process NMSE reaches 0.18 with respect to the ground truth. We are not able to collect samples around borders due to physical limitations of the robotic platform.

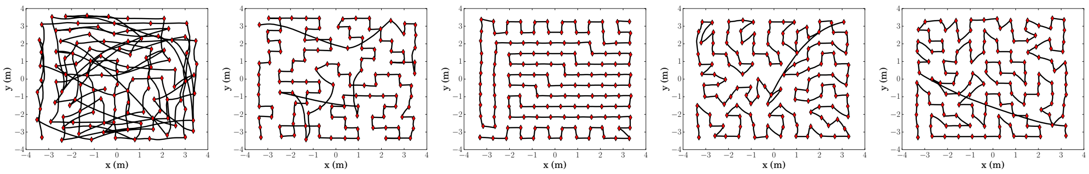
As a remark, the SS benchmark is skipped for the obstacle-free scenario since there are no frontiers. Moreover, the map is explored already in the first step since perception sensor radius is larger than the environment size.

S1 simulation results. In Table 5 we observe that our IE strategy outperforms benchmarks with respect to the total travelled distance. While GGE collects the least amount of process samples $N_{\text{proc.}}$ while reaching the same NMSE (making it computationally less expensive), it produces significantly longer trajectories. As a reminder, with GRBCM we can compensate computation costs introduced due to a higher number of process samples, thus making the total distance travelled our main comparison tool between different strategies.

Looking into the example of the resultant trajectories (Figure 8) in the obstacle-free scenario (equivalent to exploring one big empty room), the behavior of GGE (Figure 8a) is stepping towards the highest value of $H(\mathbf{x})$ while skipping some close locations with $H(\mathbf{x}) > \epsilon$, and forcing it in later stages to traverse longer distances. While GLGE (Figure 8b) strategy avoids initially traversing larger distances, it lacks a systematic approach and still misses multiple informative locations, making exploration more "expensive" in the final stages when the environment is fully observed, but the process itself is not yet fully explored in many previously visited local areas. If the room size is covered by search radius r in IE (Figure 8c), we can solve it efficiently within one multi-step exploration sequence, avoiding any expensive returns to the vicinity of previously explored locations. However, we can observe that even with a smaller radius we still systematically explore the horizon r before moving forward (Figures 8d,e), thus behaving likewise to GLGE only in cases when the local area is completely explored. Consequently, our strategy produces significantly shorter trajectories.

Table 5. Mean value and standard deviation over five runs for scenario S1. Process exploration: Total travelled distance and number of collected process measurements ($N_{\text{proc.}}$) required to achieve NMSE of 0.18 with respect to the ground truth.

Strategy	Distance (m)	$N_{\text{proc.}}$
GGE	140.18 ± 13.36	109 ± 2
GLGE	107.06 ± 2.12	134 ± 3
IE $r = 2$ m	102.82 ± 3.51	134 ± 3
IE $r = 3$ m	105.39 ± 1.84	134 ± 2
IE $r =$ full	104.49 ± 2.79	138 ± 4



(**a**) GGE (**b**) GLGE (**c**) IE $r =$ full (**d**) IE $r =$ 2m (**e**) IE $r =$ 3m

Figure 8. Example trajectory for scenario S1 and starting location (−3.3,−3.3) for the different strategies.

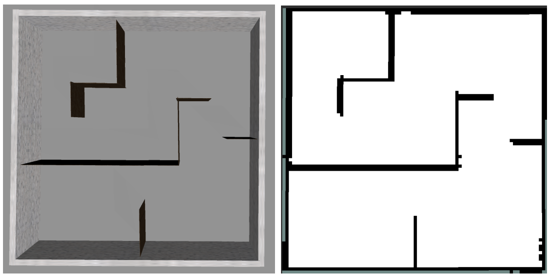
Small room-like environment, obstacles introduced (S2 in Table 1). In this evaluation, we assume that the process is explored when NMSE reaches 0.21 with respect to the ground truth. As before, we are not able to collect samples in close proximity of obstacles.

S2 simulation results. When we add obstacles, we observe that IE tends to produce lower process error for the same map coverage (Table 6). We also see that our strategy collects more samples and traverses larger distances before the map is fully explored. This is due to its focus on exploring the surrounding process before continuing further towards unexplored areas. The compared benchmarks favor moving faster towards regions with high process uncertainty while ignoring some less uncertain but still important areas around its current measurement location. As a consequence of exploring room by room before moving further, IE also produces shorter final trajectories (Table 6).

Table 6. Mean value and standard deviation over five runs for scenario S2: (i) mapping: Total travelled distance needed to explore the map with accompanied number of obtained process measurements $N_{\text{proc.,map}}$ and process NMSE; (ii) process exploration: Total exploration distance needed to explore the process and number of collected measurements ($N_{\text{proc.}}$) required to achieve NMSE of 0.21 with respect to the ground truth.

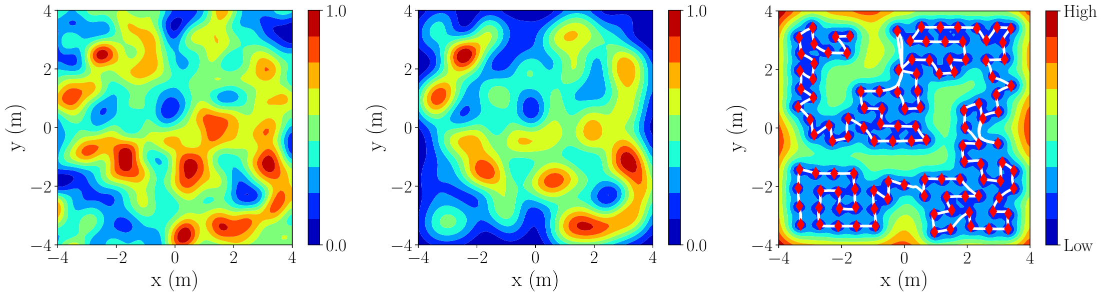
Strategy	Distance (m) Map Explored	$N_{\text{Proc.Map}}$	Proc. NMSE Map Explored	Distance (m) Proc. Explored	$N_{\text{proc.}}$
GGE	15.84 ± 4.38	13 ± 5	0.6 ± 0.1	126.31 ± 5.13	84 ± 1
GLGE	49.77 ± 10.72	64 ± 12	0.48 ± 0.05	94.54 ± 2.56	114 ± 4
SS	25.72 ± 6.63	35 ± 9	0.62 ± 0.02	108.17 ± 7.38	118 ± 4
IE $r = 2$ m	69.94 ± 12.58	89 ± 15	0.42 ± 0.07	89.89 ± 4.97	114 ± 2
IE $r = 3$ m	64.89 ± 17.68	85 ± 17	0.47 ± 0.12	90.97 ± 3.96	116 ± 2

An example of IE mapping for the starting position $(x(\text{m}), y\ (\text{m})) = (-3.3, -3.3))$ with $r = 3$ m is shown below (Figure 9). For this setup, we see the results of the reconstructed process $\mathbf{y}_*$ (based on Process 1—Figure 10a), represented by mean (Figure 10b) and variance (Figure 10c).



(**a**) Small room-like (**b**) $\mathcal{M}$

Figure 9. Environment for scenario S2.



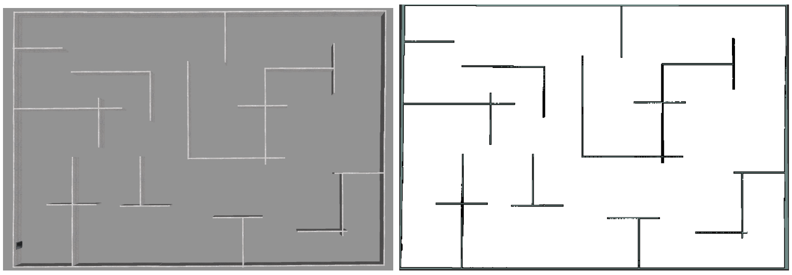
(**a**) **y** (**b**) μ_* (**c**) $\mathbf{\Sigma}_*$

Figure 10. S2: IE $r = 3m$, $k = 3l$ with red diamonds in (c) representing measurements and white lines trajectory.

Large room-like environment, obstacles introduced (S3 in Table 1). In addition to the multiple tests performed with S1 and S2, we test our strategy against GLGE in a large scale environment covering a surface of 2154.24 m^2 (Figure 11). We assume that the process is explored when process NMSE reaches 0.084 with respect to the ground truth.

S3 simulation results. We note a lower final NMSE error in S3, in comparison with S1/S2. This is due to higher process correlation in process 2, resulting in better estimation of unreachable areas $\{\mathcal{X} \setminus \mathcal{X}_{\text{free}}\}$. The estimated process for $r = 3$ m. is shown in Figure 12.

Despite our IE strategy ($r = [3, 6]$ m) requiring more measurements to reach the same NMSE as GLGE (Table 7), GLGE produces more than a 100 m longer trajectory for exploring the full environment, making our strategy more efficient regarding distance measure.



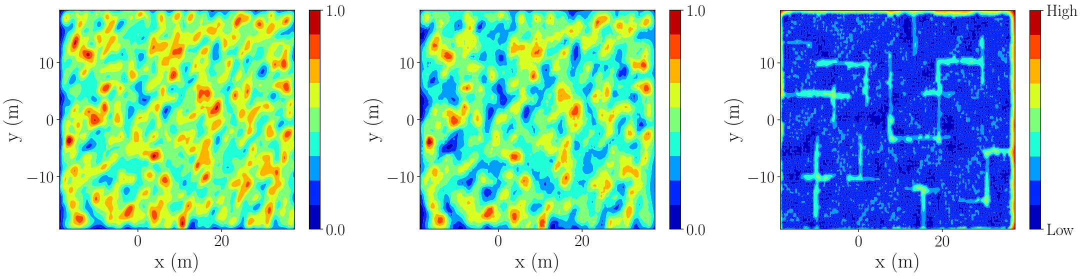
(**a**) Large room-like (**b**) $\mathcal{M}$

Figure 11. Environment for scenario S3.

The reason for GLGE placing fewer measuring points lies in its greedy movement strategy, as it always places only one next best point to satisfy Equation (8). In contrast, our strategy places as many points as possible within the minimum sampling distance that are above the threshold ϵ (Algorithm 1). As a trade-off, more points are generated, but a smaller distance is needed to visit all points. The executed trajectories are shown in Figure 13.

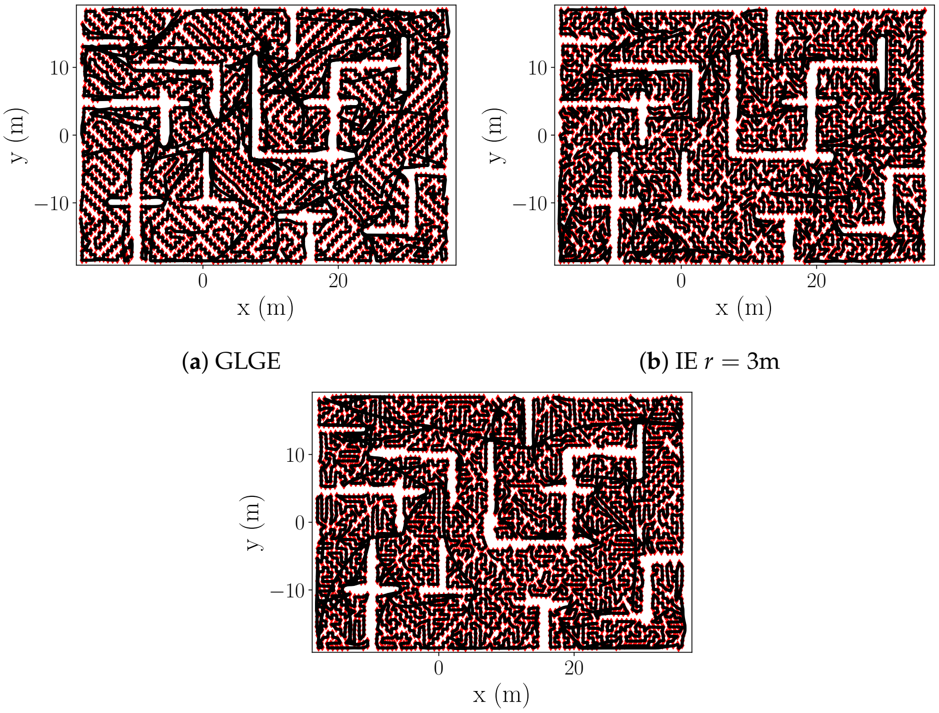
Table 7. Scenario S3: (i) mapping: Total travelled distance needed to explore the map with accompanied number of obtained process measurements $N_{\text{proc.,map}}$ and process NMSE; (ii) process exploration: Total exploration distance to explore the process and number of collected measurements ($N_{\text{proc.}}$) required to achieve NMSE of 0.084 with respect to the ground truth.

Strategy	Distance (m) Map Explored	$N_{\text{Proc.Map}}$	Proc. NMSE Map Explored	Distance (m) Proc. Explored	$N_{\text{proc.}}$
GLGE	1398.45	2027	0.29	2977.52	3228
IE $r = 3$ m	1393.46	1852	0.33	2847.06	3423
IE $r = 6$ m	1888.78	2530	0.32	2872.52	3567



(**a**) **y** (**b**) $\boldsymbol{\mu}_*$ (**c**) $\boldsymbol{\Sigma}_*$

Figure 12. S3: IE $r = 3m, k \ = 3l$.



(**a**) GLGE (**b**) IE $r = 3$m

(**c**) IE $r = 6$m

Figure 13. S3: Trajectories (black lines) and obtained measurements (red markers).

As a final remark, we consider a different final NMSE for each scenario for stating the process as fully explored. Given the same minimum sampling distance k, there are two main underlying reasons for that: (i) different length-scales l for the processes where higher length-scales allow us to better estimate unobservable areas below obstacles; and (ii) various density of the obstacles within operating environment resulting in diversity of unreachable regions.

6. Experiments

The experiments are performed in order to verify that the system is operating successfully under laboratory conditions.

6.1. Experimental Setup

The experiment (E1 in Table 1) is conducted with the slider (Figure 2a) in "Holodeck" laboratory, populated with obstacles (Figure 14a). All computations are carried out in real-time with the on-board computer of the robot. The aim is to reconstruct the magnetic intensity field of the "Holodeck". As mentioned before (Section 4.1), we use real-world magnetic-field data with a simulated process sensor in the experiment.

First, the algorithm performance is tested with two different sampling distances for IE strategy:

- Finite horizon, $r = [2, 3]$ m.
- Sampling distance $k = [2l, 3l]$.

Additionally, IE with $k = 3l$ is compared against the aforementioned baselines (Section 5.3.1).

Robot operates in an environment of size 6×12 m^2, while the magnetic field data was collected within 4.6×9.6 m^2 (Table 1). Hyper-parameters of magnetic field intensity are $\boldsymbol{\theta} = [0.07, 0.20, 1.41 \times 10^{-4}]^T$ (Table 2). All runs start from $\mathbf{x}^{[0]} = [0, 0]^T$ (the center of the operating environment).

As a remark, we can notice that in the real-experiment we have a bigger operating environment than process exploration area. This is due to a priori collected magnetic field measurements from partial environment. However, we allow path planner to plan through the whole operation environment while $\mathcal{X}_{\text{poi}}$ are generated only within the exploration area. This presents a possibility to collect data closer to exploration borders.

6.2. Experimental Results

As expected, based on previously obtained simulation results (Section 5.2), in order to reach the same NMSE with respect to the ground truth, a smaller sampling size results in (i) longer trajectories, i.e., longer exploration time, and (ii) more samples obtained, i.e., more computational costs (Table 8).

Table 8. Experiment E1—variation of sampling size: (i) mapping: Total travelled distance needed to explore the map with accompanied number of obtained process measurements up to that point $N_{\text{proc.,map}}$ and process NMSE; (ii) process exploration: Total exploration distance to explore the process and number of collected measurements ($N_{\text{proc.}}$) required to achieve NMSE of 0.34 with respect to ground truth.

Strategy	Distance (m) Proc. Explored	$N_{\text{proc.}}$	Distance (m) Proc. Explored	$N_{\text{proc.}}$
	$k = 2l$		$k = 3l$	
IE $r = 2$ m	92.84	144	55.63	75
IE $r = 3$ m	93.56	146	58.23	74

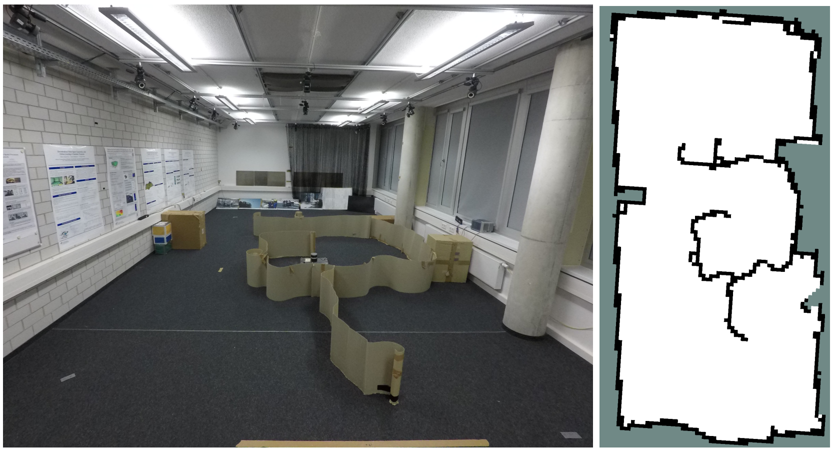
Real robot has its aforementioned physical limitations (here we set a safety radius of 0.6 m from the center of the robot) resulting in a higher NMSE for the explored process since we cannot measure in proximity and below obstacles. In evaluation, we assume that the process is fully explored when the NMSE is 0.34 with respect to the ground truth.

Based on the comparison of results presented in Table 9, we see that our systematic exploration (IE with horizon $r = [2, 3]$ m) results in significantly smaller distances needed for reaching NMSE 0.34. GGE (Figure 15a) strategy results in backward-forward motions. Both GGE and SS (Figure 15c) are suitable for fast map exploration, with the cost of notably slower process exploration. While GLGE (Figure 15b) behaves similarly with IE in narrow corridors, it lacks systematic approach within wider areas which produces longer trajectories due to returning to the vicinity of previously explored regions. We also see that with a small radius with respect to the rooms/corridors in the environment, our strategy (IE $r = 2$ m, Figure 15d) can still miss a few important sampling locations and needs to traverse back. Nevertheless, both IE strategies (Figure 15d,e) result in shorter and more systematic trajectories.

Figure 14b displays an example of the reconstructed occupancy grid map $\mathcal{M}$ and Figure 16b,c show the estimated process $\mathbf{y}_*$ by the IE ($r = 3$ m) in the real-world experiment for real magnetic field intensity of Figure 16a.

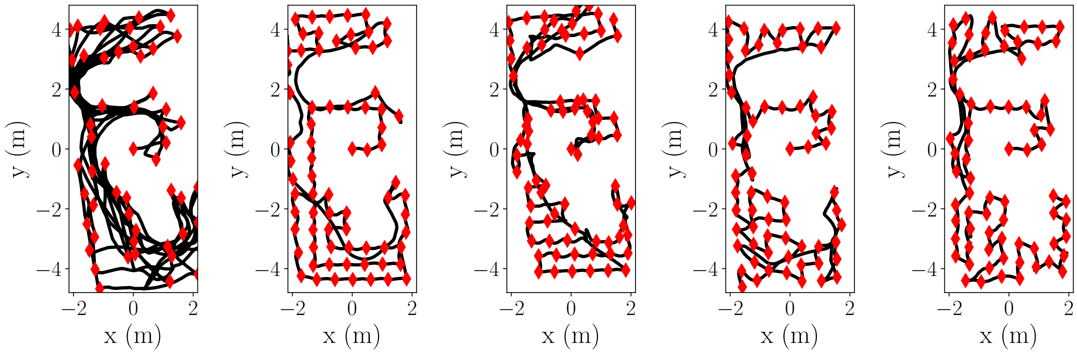
Table 9. Experiment E1: (i) mapping: Total travelled distance needed to explore the map with accompanied number of obtained process measurements up to that point $N_{\text{proc.,map}}$ and process NMSE; (ii) process exploration: Total exploration distance to explore the process and number of collected measurements ($N_{\text{proc.}}$) required to achieve NMSE of 0.34 with respect to ground truth.

Strategy	Distance (m) Map Explored	$N_{\text{proc.map}}$	Proc. NMSE Map Explored	Distance (m) Proc. Explored	$N_{\text{proc.}}$
GGE	17.28	17	0.71	170.94	53
GLGE	44.91	54	0.62	64.81	75
SS	33.24	48	0.68	73.36	77
IE $r = 2$ m	47.45	62	0.54	55.62	75
IE $r = 3$ m	53.08	66	0.53	58.23	74



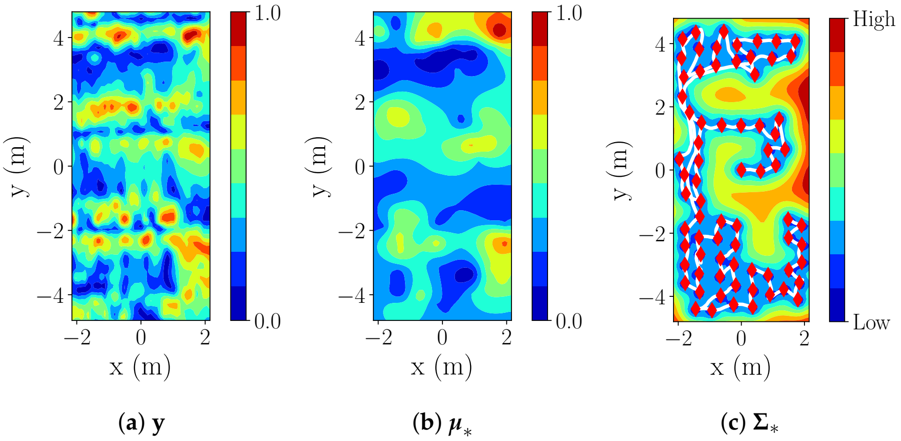
(**a**) Holodeck (**b**) $\mathcal{M}$

Figure 14. E1: Experiment with holonomic robotic platform.



(**a**) GGE (**b**) GLGE (**c**) SS (**d**) IE $r = 2$m (**e**) IE $r = 3$m

Figure 15. E1: Trajectories comparison, $k = 3l$.



(a) **y** **(b)** $\boldsymbol{\mu}_*$ **(c)** $\boldsymbol{\Sigma}_*$

Figure 16. E1: IE, $r = 3m$, $k = 3l$ with red diamonds in (c) representing measurements and white lines trajectory.

7. Conclusions and Future Work

This paper focuses on the exploration of a spatial process in an unknown environment with an autonomous mobile robot. More precisely, we address a scenario where two fundamental assumptions that are typically present in the state of the art do not hold: (i) A priori knowledge about the operating environment, and (ii) that process and map exploration can be solved independently. For addressing this, we propose an integrated strategy that solves the process and map exploration jointly by making the exploration strategy systematic. Meaning, it first explores local area before proceeding further towards unexplored areas.

Our approach has two essential components: (i) Selection of candidate points to decrease process uncertainty and increase map coverage, (ii) ordering of those points to minimize total travelled distance while incorporating knowledge about undiscovered map regions. We intertwine these two components by building on frontiers-based and GPs-entropy-driven exploration, A* path planner, and solutions to VRP.

We evaluated our integrated strategy in both simulations and experiments, using a simulated GP and magnetic field intensity as processes at hand, in various environments. The results allowed us to conclude the following: (i) Our strategy is able to decrease the error between process estimate (as calculated with GPs) and ground truth faster than GPs-greedy-entropy-based and frontiers-based strategies; (ii) trajectories resulting from our strategy follow a structured path, in contrast to the oscillatory behavior of GPs-entropy-driven exploration; and (iii) our integrated approach offers the best trade-off between process estimation error and map coverage.

To conclude, integrated exploration offers an opportunity to explore unknown environments efficiently, especially benefitting from segment exploration, e.g., exploring larger rooms or wide corridors in detail before moving towards “exits” to unknown space, controlled by the horizon parameter r and minimal sampling distance k. We note that expectantly choosing k closer to the length-scale l produces shorter trajectories, while sacrificing precision of process estimation.

Let us finally point out that in this work our focus was primarily on the development and testing of the exploration concept; exploitation of SLAM loop-closures was not in the scope of this paper. This is left for a future extension of the proposed system. Therefore, further efforts shall be devoted to a full integration of pose uncertainties propagated from SLAM into the GPs. Additionally, we aim to extend our exploration concept by incorporating an active SLAM component for an efficient trade-off between exploration–exploitation of both exploration and robot localization.

Author Contributions: Investigation, V.K.; Methodology, V.K., A.V., D.S., and L.M., Software, V.K.; Supervision, A.V., L.M. and D.S.; Validation, V.K.; Writing—original draft, V.K., A.V. and L.M.; Writing—review & editing, V.K., A.V., L.M. and D.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research of V.K. was funded by DLR-DAAD Research Fellowship. The work of L.M. was partially funded by the Spanish Ministry of Science, Innovation and Universities Project COMCISE (RTI2018-100847-B-C22, MCIU, AEI, FEDER, UE).

Acknowledgments: The Institute of Communications and Navigation (DLR) provided technical support and materials used for experiment.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GP	Gaussian process
NBC	nuclear, biological, chemical
SLAM	Simultaneous Localization and Mapping
BCM	Bayesian Committee Machine
RBCM	Robust Bayesian Committee Machine
TSP	Traveling Salesman Problem
FOV	field of view
GRBCM	Generalized Robust Bayesian Committee Machine
SE	squared exponential
LIDAR	Light Detection and Ranging
POI	Point of Interest
ICP	Iterative Closest Point
TEB	Time Elastic Band
WFD	Wavefront Frontier Detector
BFS	Breadth-first search
VRP	Vehicle Routing Problem
IE	Integrated exploration
NMSE	Normalized Mean Square Error
GGE	Greedy global entropy
GLGE	Greedy local-global entropy
SS	Sequential strategy
ROS	Robot Operating System

References

1. He, X.; Bourne, J.R.; Steiner, J.A.; Mortensen, C.; Hoffman, K.C.; Dudley, C.J.; Rogers, B.; Cropek, D.M.; Leang, K.K. Autonomous chemical-sensing aerial robot for urban/suburban environmental monitoring. *IEEE Syst. J.* **2019**, *13*, 3524–3535. [CrossRef]
2. Bird, B.; Griffiths, A.; Martin, H.; Codres, E.; Jones, J.; Stancu, A.; Lennox, B.; Watson, S.; Poteau, X. A robot to monitor nuclear facilities: Using autonomous radiation-monitoring assistance to reduce risk and cost. *IEEE Robot. Autom. Mag.* **2018**, *26*, 35–43. [CrossRef]
3. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
4. Carrillo, H.; Reid, I.; Castellanos, J.A. On the comparison of uncertainty criteria for active SLAM. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 2080–2087.
5. Valencia, R.; Andrade-Cetto, J. Active pose SLAM. In *Mapping, Planning and Exploration with Pose SLAM*; Springer: Berlin, Germany, 2018; pp. 89–108.
6. Yamauchi, B. A frontier-based approach for autonomous exploration. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA'97), Monterey, CA, USA, 10–11 July 1997; Volume 97, pp. 146–151.

7. Bourgault, F.; Makarenko, A.A.; Williams, S.B.; Grocholsky, B.; Durrant-Whyte, H.F. Information based adaptive robotic exploration. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; Volume 1, pp. 540–545.
8. Viseras, A.; Shutin, D.; Merino, L. Robotic active information gathering for spatial field reconstruction with rapidly-exploring random trees and online learning of Gaussian processes. *Sensors* **2019**, *19*, 1016. [CrossRef] [PubMed]
9. Rasmussen, C.E.; Williams, C.K. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*; The MIT Press: Cambridge, MA, USA, 2005.
10. Krause, A.; Singh, A.; Guestrin, C. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.* **2008**, *9*, 235–284.
11. Singh, A.; Ramos, F.; Whyte, H.D.; Kaiser, W.J. Modeling and decision making in spatio-temporal processes for environmental surveillance. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 4–8 May 2010; pp. 5490–5497.
12. Julian, B.J.; Angermann, M.; Schwager, M.; Rus, D. Distributed robotic sensor networks: An information-theoretic approach. *Int. J. Robot. Res.* **2012**, *31*, 1134–1154. [CrossRef]
13. Marchant, R.; Ramos, F. Bayesian Optimisation for informative continuous path planning. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6136–6143.
14. Fink, J.; Kumar, V. Online methods for radio signal mapping with mobile robots. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–8 May 2010; pp. 1940–1945.
15. Carrillo, H.; Dames, P.; Kumar, V.; Castellanos, J.A. Autonomous robotic exploration using a utility function based on Rényi's general theory of entropy. *Auton. Robot.* **2018**, *42*, 235–256. [CrossRef]
16. Elfes, A. Using occupancy grids for mobile robot perception and navigation. *Computer* **1989**, *22*, 46–57. [CrossRef]
17. Ghaffari Jadidi, M.; Valls Miro, J.; Valencia, R.; Andrade-Cetto, J. Exploration on continuous Gaussian process frontier maps. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6077–6082.
18. Lilienthal, A.J.; Loutfi, A.; Blanco, J.L.; Galindo, C.; Gonzalez, J. Integrating SLAM into gas distribution mapping. In Proceedings of the ICRA Workshop on Robotic Olfaction, Towards Real Applications (ICRA), Rome, Italy, 10 April–14 April 2007; pp. 21–28.
19. Jung, J.; Oh, T.; Myung, H. Magnetic field constraints and sequence-based matching for indoor pose graph SLAM. *Robot. Auton. Syst.* **2015**, *70*, 92–105. [CrossRef]
20. Prágr, M.; Čížek, P.; Bayer, J.; Faigl, J. Online incremental learning of the terrain traversal cost in autonomous exploration. In Proceedings of the Robotics: Science and Systems 2019, Freiburg im Breisgau, Germany, 22–26 June 2019.
21. Kulich, M.; Faigl, J.; Přeučil, L. On distance utility in the exploration task. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 4455–4460.
22. Oßwald, S.; Bennewitz, M.; Burgard, W.; Stachniss, C. Speeding-up robot exploration by exploiting background information. *IEEE Robot. Autom. Lett.* **2016**, *1*, 716–723.
23. Kulich, M.; Kubalík, J.; Přeučil, L. An integrated approach to goal selection in mobile robot exploration. *Sensors* **2019**, *19*, 1400. [CrossRef] [PubMed]
24. Basilico, N.; Amigoni, F. Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. *Auton. Robot.* **2011**, *31*, 401. [CrossRef]
25. Amigoni, F.; Gallo, A. A multi-objective exploration strategy for mobile robots. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 3850–3855.
26. Calisi, D.; Farinelli, A.; Iocchi, L.; Nardi, D. Multi-objective exploration and search for autonomous rescue robots. *J. Field Robot.* **2007**, *24*, 763–777. [CrossRef]
27. Stachniss, C.; Burgard, W. Exploring unknown environments with mobile robots using coverage maps. In Proceedings of the IJCAI, Acapulco, Mexico, 9–15 August 2003; Volume 2003, pp. 1127–1134.
28. Choquet, G. Theory of capacities. *Annales de l'institut Fourier* **1954**, *5*, 131–295. [CrossRef]

29. Liu, H.; Cai, J.; Wang, Y.; Ong, Y.S. Generalized robust bayesian committee machine for large-scale Gaussian process regression. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 3137–3146.
30. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [CrossRef]
31. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 21 June–18 July 1965; Volume 1, pp. 281–297.
32. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
33. Pomerleau, F.; Colas, F.; Siegwart, R.; Magnenat, S. Comparing ICP Variants on Real-World Data Sets. *Auton. Robot.* **2013**, *34*, 133–148. [CrossRef]
34. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014; Volume 2, p. 9.
35. Shan, T.; Englot, B. LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
36. Dubé, R.; Cramariuc, A.; Dugas, D.; Nieto, J.; Siegwart, R.; Cadena, C. SegMap: 3D segment mapping using data-driven descriptors. In Proceedings of the Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, 26–30 June 2018.
37. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
38. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
39. Rösmann, C.; Hoffmann, F.; Bertram, T. Integrated online trajectory planning and optimization in distinctive topologies. *Robot. Auton. Syst.* **2017**, *88*, 142–153. [CrossRef]
40. Keidar, M.; Kaminka, G.A. Efficient frontier detection for robot exploration. *Int. J. Robot. Res.* **2014**, *33*, 215–236. [CrossRef]
41. Toth, P.; Vigo, D. *The Vehicle Routing Problem*; SIAM: Philadelphia, PA, USA, 2002; p. 367. [CrossRef]
42. GPy. GPy: A Gaussian Process Framework in Python. Available online: http://github.com/SheffieldML/GPy (accessed on 18 June 2019).
43. Van Omme, N.; Perron, L.; Furnon, V. *Or-Tools User's Manual*; Technical Report; Google: Mountain View, CA, USA, 2014.
44. Viseras-Ruiz, A.; Wiedemann, T.; Manss, C.; Magel, L.; Carsten-Mueller, J.; Shutin, D.; Merino, L. Decentralized multi-agent exploration with online-learning of Gaussian processes. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4222–4229.