

Online Incremental Learning of the Terrain Traversal Cost in Autonomous Exploration

Miloš Prágr

Petr Čížek

Jan Bayer

Jan Faigl

Abstract—In this paper, we address motion efficiency in autonomous robot exploration with multi-legged walking robots that can traverse rough terrains at the cost of lower efficiency and greater body vibration. We propose a robotic system for online and incremental learning of the terrain traversal cost that is immediately utilized to reason about next navigational goals in building spatial model of the robot surrounding. The traversal cost experienced by the robot is characterized by incrementally constructed Gaussian Processes using Bayesian Committee Machine. During the exploration, the robot builds the spatial terrain model, marks untraversable areas, and leverages the Gaussian Process predictive variance to decide whether to improve the spatial model or decrease the uncertainty of the terrain traversal cost. The feasibility of the proposed approach has been experimentally verified in a fully autonomous deployment with a hexapod walking robot.

I. INTRODUCTION

Multi-legged walking robots are capable of rough terrains traversal, either by leveraging detailed foothold position plans [2, 37], or reactively utilizing tactile information [4, 10]. On the other hand, the robots may suffer from poor energy efficiency [13] and low stability [18]; hence, they can benefit from traversal cost prediction of the observed terrains. In unknown environments, the robot may encounter previously unobserved terrain types, and therefore, it needs to explore and actively update its terrain traversal cost model to improve its performance as the perception is active by nature [1].

We propose to address autonomous robotic exploration as a problem to simultaneously create a spatial model of the unknown environment together with incremental learning of the traversal cost model. The spatial model is employed to reason about untraversable areas, but incrementally learned traversal cost characterizes the robot experience with its locomotion effectiveness over traversable terrains. Thus, the learned model is employed in the extrapolation of the traversal cost assessment to observed but not yet visited areas, to intentionally avoid hard-to-traverse areas.

The spatial frontier-based exploration [35] can be utilized to navigate the robot towards passable areas at the boundary of the explored space. However, there is not an easily distinguishable boundary in exploring some underlying phenomena such as the terrain traversal cost. Therefore, model confidence can

Authors are with the Czech Technical University, Faculty of Electrical Engineering, Technická 2, 166 27, Prague, Czech Republic {pragrmil|cizekpe6|bayerjal|faigl.j}@fel.cvut.cz

The presented work has been supported by the Czech Science Foundation (GACR) under research project 18-18858S. The authors acknowledge the support of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 “Research Center for Informatics”.

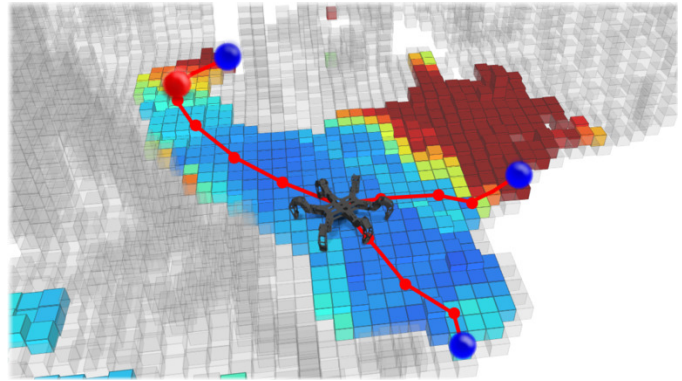


Fig. 1. Visualization of reasoning about possible navigational goals in the spatial frontier-based (blue spheres) exploration combined with building terrain traversal cost model (jet color coding). The robot selects to actively improve its traversal cost model for areas with low model confidence (red spheres). The model is instantly utilized in path planning, and thus the robot avoids areas that are believed to be hard-to-traverse.

be used to reason about active navigation towards areas with low model fidelity [29, 34]. An example of such reasoning within the proposed approach is visualized in Fig. 1.

The confidence of the terrain traversal cost model can be obtained using Gaussian Process (GP) regression, a nonparametric generalization of the linear regression which can extrapolate both the predictive mean and variance. The GP regression has been used in spatial exploration with a continuous spatial occupancy [20, 26] or terrain elevation models [32]. However, the GP regression suffers from cubic learning time complexity, and the Bayesian Committee Machine (BCM) with the GP regressors in frontier-based exploration [14] has been utilized to create spatial occupancy representation [33].

Motivated by recent advancements on the GPs and BCM, we propose to use the Robust BCM (RBCM) [8] for incremental construction of the terrain traversal cost model in the exploration of unknown environments. The robot thus alters navigation towards frontiers of the spatial exploration and areas with low confidence of the traversal cost model that is characterized by a high predictive variance; while simultaneously exploiting the learned model to avoid costly terrains.

In this paper, we describe the developed robotic system that represents an integrated framework with complete pipeline of sensing, model building, informed planning, and execution that has been experimentally verified in autonomous experiments. Regarding the existing work, we consider the main contributions of the presented approach as follows.

- Robotic system with active improvements of the terrain traversal cost model deployed in the autonomous exploration of the spatial model and the traversal cost model.
- Experimental validation of the proposed system in autonomous exploration with the hexapod walking robot.
- Deployment of online incremental learning of the underlying traversal cost model using the RBCM with GP regressor experts over the terrain feature descriptor space.
- Experimental evaluation of (fast) incremental learning approaches within the addressed terrain traversal cost modeling task.

The rest of the paper is structured as follows. Related approaches on terrain traversal cost model and characterization are overviewed in the following section. A brief description of the used RBCM is presented in Section III. The main parts of the proposed framework are described in Section IV and results on its experimental validation are reported in Section V. The paper is concluded in Section VI.

II. OVERVIEW OF THE EXISTING TERRAIN TRAVERSAL COST AND TERRAIN CHARACTERIZATION APPROACHES

Robots autonomously navigating in rough terrains must identify and avoid risks such as possible robot damage or energy wasting due to low efficiency. The risk and efficiency of the terrain traversal can be defined by characterizing remotely observed terrains or by examining the robot experience of the actual traversal. Observed geometrical and appearance properties of the perceived environment can be used for a remote characterization but the traversal cost defined as the level of risk and locomotion efficiency needs to be based on the robot traversal experience with the terrain.

Geometric properties such as height [13, 28], slope [6], or roughness [16] are directly connected to the viability of the terrain traversal. Multiple geometric properties can be used to detect unpassable areas and describe safe terrains, e.g., by a combined danger index in [28]. The appearance descriptors leverage the frequency domain or the colors of the observed areas. In [27], Gabor filters are used to describe overhead imagery, while voxel color information is directly utilized in [3], and color and reflectance vegetation indices are reviewed in [36]. Appearance and geometric descriptors may not only directly define the terrain traversal feasibility but can also be used to learn alternative terrain characterizations such as the terrain classification and robot experience.

Terrain classification is a task to assess the terrain into a set of discrete terrain classes based on human labeled terrain types [21] or to cluster unlabeled terrains [11]. Individual terrain types can carry information about the terrain traversability, e.g., an unpassable obstacle class [5]. Terrain classification can be based on geometrical and appearance properties of remotely observed terrains [16] but also on proprioceptive sensing [11]. A combination of the appearance and vibration terrain characterization is, for example, utilized in [19, 21].

Experience with the terrain traversal can be characterized as the observed difficulty of the robot with walking over the traversed terrain, and it can encode the tradeoff between

various measures of the traversal efficiency. The Static Stability Margin [18] and the Dynamic Stability Margin [17] measure the stability of the multi-legged robot by observing its support polygon, i.e., the polygon defined by its footholds, and projection of the center of gravity. The concept of robot stability is also related to the vehicle vibration, which may decrease the perception accuracy, and eventually damage the robot construction. Alternatively, the terrain traversal experience can be encoded in a performance measure such as the cost of transport that is defined as the ratio between the consumed energy and velocity of the robot [15, 31].

The proposed framework (described in Section IV) is tailored to employ any continuous experience-based performance measures, but for our herein presented particular setup with hexapod walking robot, the terrain traversal risk is characterized as the experienced stability of the robot over a predefined time window. However, such a traversal risk is a particular instance for the system deployment, and thus its description is dedicated to Section V.

III. GAUSSIAN PROCESS REGRESSION

A brief overview of the Robust Bayesian Committee Machine (RBCM) inference mechanism is presented here together with a summary of the GP regression and BCM with GP regressor experts to make the paper more self-contained.

GP regression is a non-parametric generalization of the linear regression and for given noisily observed function $f(x)$

$$y = f(x) + \epsilon, \quad \epsilon \in \mathcal{N}(0, \sigma^2), \quad (1)$$

the GP is defined as a distribution over functions [25]

$$f(x) \sim \mathcal{GP}(m(x), K(x, x')) \quad (2)$$

characterized by its mean $m(x)$ and covariance $K(x, x')$ as

$$m(x) = E[f(x)], \quad (3)$$

$$K(x, x') = E[(f(x) - m(x))(f(x') - m(x'))], \quad (4)$$

for any pair of (x, x') out of the input space \mathcal{X} . Given the training data $(X, y) = \{X_i, y_i\}_{i=1}^n$ with the size n , the predictive equations of the latent values f_* for the test data X_* can be determined as

$$\begin{aligned} \mu(X_*) &= K_* [K + \sigma^2 I]^{-1} y, \\ (\sigma(X_*))^2 &= K_{**} - K_*^T [K + \sigma^2 I]^{-1} K_*, \end{aligned} \quad (5)$$

where the notation for $(\sigma(X_*))^2$ is abused to improve the readability and clarity of the equation, and K, K_* , and K_{**} are covariance matrices defined as

$$K = K(X, X), K_* = K(X, X_*), K_{**} = K(X_*, X_*). \quad (6)$$

The GP regression has cubic learning time complexity $\mathcal{O}(n^3)$, which limits its application in tasks such as the robotic exploration, and therefore, less demanding approach is desirable.

The BCM [30] is a product of experts approach that allows combining GP regressors learned on multiple datasets. Since each of the GP regressors can be constructed independently, the learning time complexity of the BCM with GP regressors

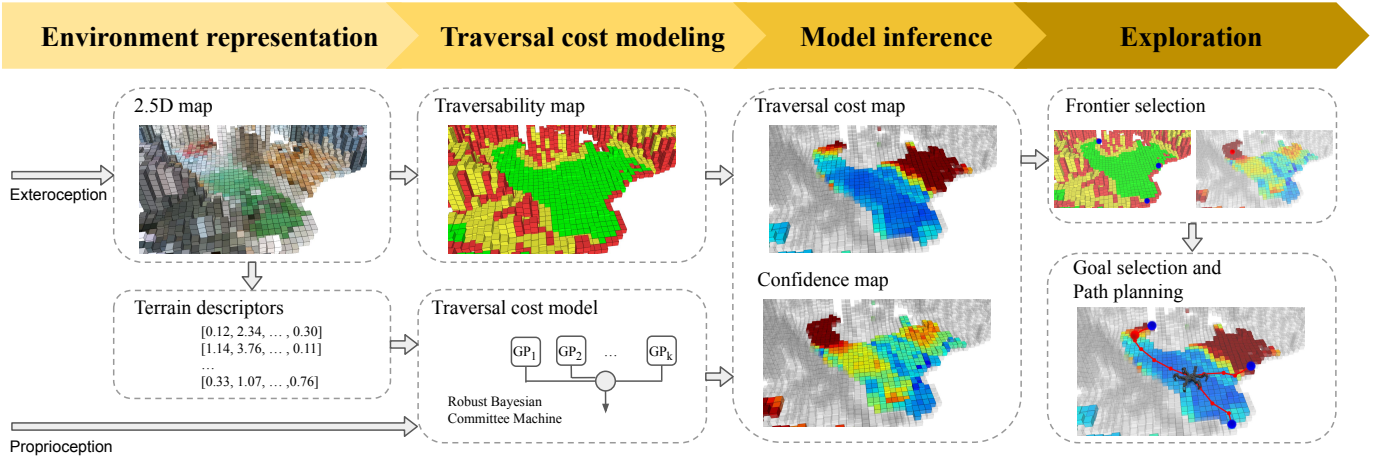


Fig. 2. Overview of the proposed system for online incremental learning of the terrain traversal cost in autonomous spatial exploration.

is $\mathcal{O}((n/k)^3k)$, where k is the number of equally sized components of the size m , i.e., $n = km$. In practice, the size of a single component m is significantly smaller than n , and thus its processing time can be considered constant. Thus for a fixed m and $m \ll k$ we can consider $\mathcal{O}(m^3k) \sim \mathcal{O}(k)$ and since k is proportional to n , the total complexity of k regressors can be approximately bounded by $\mathcal{O}(n)$, a notable improvement over the original $\mathcal{O}(n^3)$. The predictive equations of the BCM with k Gaussian regression experts can be defined as

$$\begin{aligned} \mu_{\text{BCM}}(X_*) &= (\sigma_{\text{BCM}}(X_*))^2 \sum_{i=1}^k (\sigma_i(X_*))^{-2} \mu_i(X_*), \\ (\sigma_{\text{BCM}}(X_*))^{-2} &= (1-k)(K_{**})^{-1} + \sum_{i=1}^k (\sigma_i(X_*))^{-2}, \end{aligned} \quad (7)$$

where $\mu_i(X_*)$ and $(\sigma_i(X_*))^2$ are the means and covariances of the individual experts. The BCM has been further improved as the Robust BCM (RBCM) in [8]. The RBCM weights the individual experts based on the predictive power of each expert at X_* and the RBCM predictive equations are

$$\begin{aligned} \mu_{\text{RBCM}}(X_*) &= (\sigma_{\text{RBCM}}(X_*))^2 \sum_{i=1}^k \beta_i(X_*) (\sigma_i(X_*))^{-2} \mu_i(X_*), \\ (\sigma_{\text{RBCM}}(X_*))^{-2} &= \\ &= \left(1 - \sum_{i=1}^k \beta_i(X_*)\right) (K_{**})^{-1} + \sum_{i=1}^k \beta_i(X_*) (\sigma_i(X_*))^{-2}, \end{aligned} \quad (8)$$

where the weight β_i of the expert i at X_* is defined as

$$\beta_i(X_*) = 0.5 (\log(K_{**}) - \log((\sigma_i(X_*))^2)). \quad (9)$$

Thus, $\beta_i(X_*)$ is the difference in the differential entropy between the prior $p(f_*|x_*)$ and posterior $p_i(f_*|x_*, X^i)$, where X^i is the training dataset of the i -th expert.

IV. AUTONOMOUS EXPLORATION WITH ONLINE INCREMENTAL TERRAIN TRAVERSAL COST LEARNING

In the addressed problem, we are motivated to build a fully autonomous system capable of operating in a priori unknown

environment and without prior knowledge about the traversal cost. We consider the proposed system within an autonomous exploration setup to simultaneously build a spatial model of the operational environment together with the traversal cost model that is learned incrementally to increase confidence in the cost estimation. Thus, during the exploration, the robot reasons how to improve the spatial and traversal cost models while it leverages on the experience accumulated in the incrementally learned traversal cost model in navigating towards spatial frontiers and avoiding costly terrains. The whole system consists of individual modules to build the spatial environment model, mark unpassable terrains, learn the terrain traversal cost model characterizing traversable terrains (continuously utilized in the determination of the next exploration goal), navigation to the selected goal, and locomotion control.

The overall system architecture can be divided into four main parts that are depicted in Fig. 2. Exteroceptive signals are processed in the environment representation to localize the robot and build a map of the robot surroundings together with extracting terrain shape and appearance feature descriptors that are further utilized in traversal cost model inference. The traversal cost modeling includes the incremental learning of the traversal cost model using the robot proprioceptive experience coupled with the terrain descriptors. Besides, non-traversable parts of the environment are labeled as areas with infinite traversal cost based on the geometric features of the created terrain elevation map to avoid unnecessary model-based traversability assessment of unpassable areas.

The traversal cost model is employed in model inference to build a cost map of reachable areas of the environment together with the confidence of the estimated cost that is utilized in the selection of the next exploration goal towards which the robot is navigated. The system continuously gathers new measurements and updates the current navigational goal until no exploration goal is determined. A detailed description of the individual parts of the system follows.

A. Environment Representation

The environment is represented as the colored 2.5D elevation grid map denoted $\mathcal{M}_{2.5D}$ (see Fig. 3a), which utilizes an

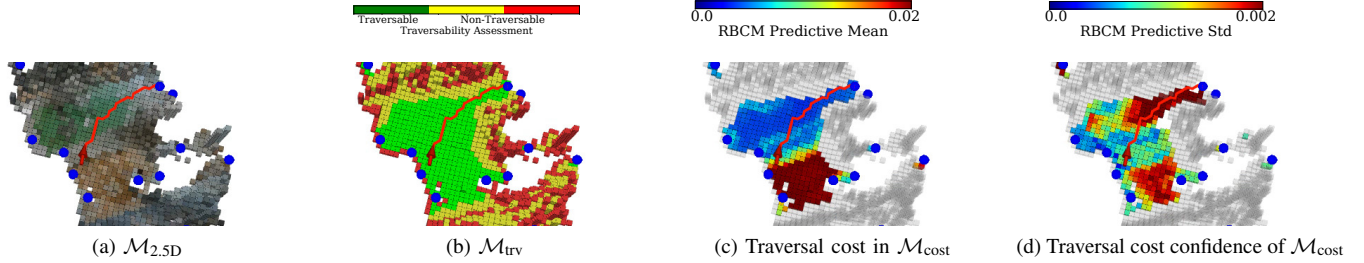


Fig. 3. Colored elevation grid map $\mathcal{M}_{2.5D}$ (a); traversability grid map \mathcal{M}_{trv} (b); traversal cost grid map \mathcal{M}_{cost} (c); and the respective confidence of the traversal cost model (d). Note, \mathcal{M}_{cost} characterizes only the traversable areas.

underlying quadtree data structure. Each cell $\nu \in \mathcal{M}_{2.5D}$ stores elevation and RGB color information, and it is further characterized with the five-dimensional geometric and appearance terrain feature descriptor $desc(\nu)$ that is a modification of the terrain descriptor used in [23]. The used geometric part of the descriptor, which is designed to distinguish the unstructured, linear, and planar shape [16] of the terrain, is defined as

$$s_1 = \frac{\lambda_1}{\lambda_3}, \quad s_2 = \frac{\lambda_2 - \lambda_1}{\lambda_3}, \quad s_3 = \frac{\lambda_3 - \lambda_2}{\lambda_3}, \quad (10)$$

where $\lambda_1 < \lambda_2 < \lambda_3$ are the eigenvalues of the covariance matrix of the elevation and spatial values in the spatial δ_{desc} -neighborhood of the cell ν . The residual sum of the squares feature utilized in [16, 23] is relaxed, and the two-dimensional appearance part of the descriptor is the δ_{desc} -neighborhood channel means of the ab channels of the Lab color space. For further information on the performance of individual descriptor parts and their combinations, we kindly refer the reader to [23].

B. Traversal Cost Modeling

The main role of the traversal cost modeling is to incrementally learn the traversal cost model based on the terrain feature descriptors. The particular cost value captures the real robot experience with the particular terrain type and it is measured by proprioceptive sensing. The model consists of the RBCM with GP regressor experts learned to allow inferring the traversal cost observed by the robot from the geometric and appearance description of the terrain. Thus, the RBCM is augmented with the terrain descriptors paired with the traversal cost experienced over the said terrain. Since the RBCM with a single expert would behave similarly to the GP; therefore, the maximal size of expert m_{max} is specified as a tradeoff between the computational requirements and achieved precision of the model. The RBCM experts are constructed incrementally, and a new expert is allocated every m_{max} observations. In particular, $m_{max} = 50$ is selected so that a new expert is allocated approx. once per minute because of the robot locomotion speed. Each expert thus consists of spatially neighboring terrains. In addition, we further specify the minimal number of observations m_{min} to consider the particular expert in the inference, because small experts may negatively spoil it with their high confidence.

In addition to the model learning, unpassable terrain areas are marked at this stage, and the $\mathcal{M}_{2.5D}$ grid map is transformed into the traversability grid map \mathcal{M}_{trv} . Inspired by [37],

we determine untraversable cells using step height defined as the maximum elevation difference of the neighboring cells. Thus, based on the motion capabilities of the particular robot, we specify the maximum h_{max} for traversable cells, and thus cells with the step height above h_{max} are considered untraversable. Besides, considering the embodiment of the robot, we further mark cells in the $\delta_{impassable}$ -radius neighborhood of such an untraversable cell also as untraversable, see Fig. 3b.

C. Model Inference

In the model inference part of the framework, the traversal cost grid map \mathcal{M}_{cost} is created combining $\mathcal{M}_{2.5D}$, \mathcal{M}_{trv} , and the current learned terrain traversal cost model. Each cell $\nu' \in \mathcal{M}_{cost}$ characterizes the inferred traversal cost accompanied with the traversal cost model confidence over the traversable regions represented by \mathcal{M}_{trv} , see Figs. 3c and 3d, respectively. In general, the resolution of \mathcal{M}_{cost} can differ from $\mathcal{M}_{2.5D}$ as the resolution affects the level of details achieved in spatial exploration, model exploration, but also path planning, and the most suitable resolution of the individual maps depends on the sensor resolution and the size and step length of the robotic platform. It might be necessary to resample maps using terrain descriptors of $\mathcal{M}_{2.5D}$ and model inference for each grid cell that is traversable according to \mathcal{M}_{trv} . Thus, for each traversable grid cell $\nu' \in \mathcal{M}_{cost}$, the closest grid cell $\nu \in \mathcal{M}_{2.5D}$ is determined and its terrain descriptor denoted $desc(\nu)$ is inferred. The model inference is employed for each traversable cell $\nu' \in \mathcal{M}_{cost}$ to estimate the cost using the traversal cost model prediction mean for the descriptor $desc(\nu)$

$$\mu(\nu') = \mu_{RBCM}(desc(\nu)), \quad (11)$$

and the model confidence is determined as the square root of the traversal cost prediction variance for $desc(\nu)$

$$\sigma(\nu') = \sigma_{RBCM}(desc(\nu)), \quad (12)$$

where high $\sigma(\nu')$ signifies low model confidence.

D. Exploration

The exploration module selects the next navigational goal location towards which the robot navigates. Different strategies to tradeoff the spatial exploration with the model improvement can be designed, but the proposed approach combines spatial frontiers and traversal cost model exploration. The employed strategy greedily improves the traversal cost by navigating to

Algorithm 1: Autonomous exploration with online incremental terrain traversal cost learning

Input: $\delta_{desc}, h_{max}, \delta_{impassable}, m_{min}, r_{spatial}, \delta_{score}, \lambda_{score}$ – Parameters of the terrain characterizing descriptors, traversability map, terrain traversal cost map and navigational goals.

```
1 repeat
2    $\mathcal{M}_{2.5D} \leftarrow \text{getSpatialModel}(\delta_{desc})$  // Using exteroception, e.g., RGB-D data
3    $\mathcal{M}_{trv} \leftarrow \text{getTraversabilityMap}(\mathcal{M}_{2.5D}, h_{max}, \delta_{impassable})$  // Mark untraversable areas using [37]
4    $\text{RBCM} \leftarrow \text{getTravelCostModel}(m_{min})$  // Get experts, each with at least  $m_{min}$  observations
5    $\mathcal{M}_{cost} \leftarrow \text{inferModel}(\text{RBCM}, \mathcal{M}_{2.5D}, \mathcal{M}_{trv})$ 
6    $\nu_{spatial} \leftarrow \text{getBestSpatialGoal}(\mathcal{M}_{2.5D}, \mathcal{M}_{cost}, \mathcal{M}_{trv}, r_{spatial})$  // Using cost (11) and (14)
7    $(\nu_{cost}, score(\nu_{cost})) \leftarrow \text{getBestCostGoal}(\mathcal{M}_{cost}, \mathcal{M}_{trv}, \delta_{score})$  // Using (12) and (15)
8    $\nu^* \leftarrow \text{selectNextGoal}(\nu_{spatial}, \nu_{cost}, score(\nu_{cost}), scoreThreshold)$  // Select the next goal using (16)
9    $path(\nu_r, \nu^*) \leftarrow \text{findPath}(\mathcal{M}_{cost}, \nu_r, \nu^*)$  // Path from the current robot position  $\nu_r$  to the goal  $\nu^*$ 
10   $\text{setNavigationWaypoints}(path(\nu_r, \nu^*))$  // Set the path as the next navigational waypoints
11 until  $\nu^*$  is  $\emptyset$ 



---


 $\text{getBestSpatialGoal}(\mathcal{M}_{2.5D}, \mathcal{M}_{cost}, \mathcal{M}_{trv}, r_{spatial})$ :
1  $\mathcal{F}_{spatial} \leftarrow \text{clusterFrontiers}(\mathcal{M}_{2.5D})$  // Cluster frontiers using (13)
2  $\mathcal{N}_{spatial} \leftarrow \text{assignReachableCells}(\mathcal{F}_{spatial}, \mathcal{M}_{cost}, \mathcal{M}_{trv}, r_{spatial})$  // Place reachable frontiers on  $\mathcal{M}_{cost}$ 
3 return  $\text{Dijkstra}(\mathcal{N}_{spatial}, \mathcal{M}_{cost})$  // Select the cheapest goal to reach using (14)



---


 $\text{getBestCostGoal}(\mathcal{M}_{cost}, \mathcal{M}_{trv}, \delta_{score})$ :
1 for all reachable  $\nu \in \mathcal{M}_{cost}$  do
2    $\Sigma \leftarrow \{\}$  // Compute score for all reachable cells
3   for all  $\nu' \in \mathcal{M}_{cost}$  where  $\|(\nu, \nu')\| < \delta_{score}$  do
4      $\Sigma \leftarrow \Sigma \cup \sigma(\nu')$  // Collect uncertainties for all cells in neighborhood using (12)
5    $score(\nu) \leftarrow \text{median}(\Sigma)$  // Compute the score using (15)
6 return  $(\text{argmax}(score), \max(score))$  // Select the reachable cell with the highest score
```

terrains that are considered unknown. If the observed terrains are sufficiently known, the robot explores the spatial frontiers.

Spatial goal locations are determined as means of clustered frontiers (representatives), where each cluster is a single connected component of the selected frontier cells. The number representative n_r of a single component is determined as [9]

$$n_r = 1 + \left\lfloor \frac{f}{D} + 0.5 \right\rfloor, \quad (13)$$

where f is the current number of frontier cells and D is the sensor range (in the number of grid cells). The set of frontier cells $\mathcal{N}_{spatial} \subset \mathcal{M}_{cost}$ is created by assigning a reachable, and thus traversable cell $\nu \in \mathcal{M}_{cost}$ that is incident with an unexplored cell. Moreover, frontier cells closer than $r_{spatial} = 0.4$ m to the current robot position ν_r are also ignored to avoid navigating to goals that the robot cannot observe en-route. The best spatial goal $\nu_{spatial}$ is selected as the representative with the lowest cost to be reached from the current robot position ν_r . The cheapest to reach spatial exploration goal is determined using Dijkstra's algorithm with the traversing cost $c(\nu_i, \nu_j)$ between two neighborhood cells $\nu_i, \nu_j \in \mathcal{M}_{cost}$ based on the cost prediction (11) and Euclidean distance $\|(\nu_i, \nu_j)\|$ between the centers of the corresponding grid cells ν_i and ν_j as

$$c(\nu_i, \nu_j) = \|(\nu_i, \nu_j)\| (\mu(\nu_i) + \mu(\nu_j)) / 2. \quad (14)$$

Dijkstra's algorithm is preferred since we need to determine the cost to reach all representatives, but the closest is selected.

The goal locations for the traversal cost model are grid cells $\nu \in \mathcal{M}_{cost}$ with a high model uncertainty that is considered as the $score(\nu)$ over δ_{score} spatial neighborhood to characterize the level of details in the model exploration. The value of $score(\nu)$ for a cell $\nu \in \mathcal{M}_{cost}$ is defined as the median of the traversal cost predictive standard deviation (12) of the neighboring cells

$$score(\nu_i) = \text{median}\{\sigma(\nu_j) \mid \|(\nu_i, \nu_j)\| < \delta_{score}\}. \quad (15)$$

The cell with the highest score is selected as the next navigational goal locations for exploring the traversal cost model.

Then, a particular exploration strategy is employed, which in our case is a preference of the model learning. Thus, the robot prefers to explore unknown terrains (if any has been observed) and the next navigational goal is the center of the grid cell ν^* selected using the decision rule

$$\nu^* = \begin{cases} \nu_{cost} & \text{if } score(\nu_{cost}) > \lambda_{score}, \\ \nu_{spatial} & \text{if a reachable spatial goal location exists,} \\ \emptyset & \text{otherwise,} \end{cases} \quad (16)$$

where λ_{score} is the threshold model confidence to do not consider further improvement of the traversal cost model. The particular value λ_{score} needs to be set to fit the traversal costs range and variance observed by the utilized robotic platform.

Finally, once the navigational goal ν^* is determined, the cheapest path is computed by Dijkstra's algorithm using (14)

and such a path is used to navigate the robot towards the goal. Since the robot collects new information during its navigation in the environment, and thus it improves its spatial and traversal cost model continuously, it is desirable to perform the decision-making at a high rate to quickly exploit new knowledge about the environment. Both the spatial model and traversal cost model are independently updated in separate execution loops defined by the maximal processing speed of the localization and model learning. The exploration loop needs to build (update) the traversability map \mathcal{M}_{trv} and infer a new traversal cost map \mathcal{M}_{cost} based on the spatial map $\mathcal{M}_{2.5D}$. Besides, the updated RBCM-based traversal cost model is employed to determine new goals and plan the paths.

A summary of the exploration loop with the individual parts of the model usage is depicted in Algorithm 1 together with the list of parameters that are specified for the particular experimental deployment in the following section.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed system for autonomous exploration with online and incremental learning of the terrain traversal cost has been experimentally validated in two scenarios. First, we compare the performance of the RBCM with GP regressor experts to pure GP regression on a dataset captured by the utilized robot. Second, we deploy the system in a fully autonomous exploration to demonstrate the online model learning and its benefit for avoiding costly terrains experienced by the robot.

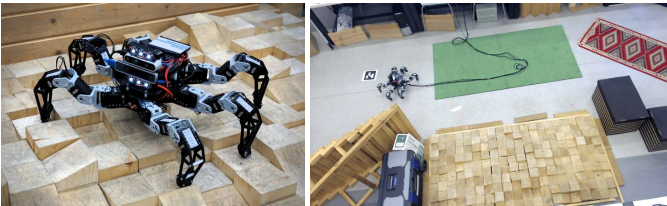


Fig. 4. The utilized hexapod walking robot and the laboratory test track.

The system is deployed on a small hexapod walking robot operating on a laboratory test track with rough terrains (see Fig. 4). The developed system consists of several modules and implementations overviewed in Table I. The substantial part of the proposed approach is the traversability cost determined from the proprioceptive measurements. Based on the review of existing work (see Section II), we chose stability based traversal cost computed as the variance of the roll angle of the sliding window spanning the robot gait cycle duration of 10 s. We found out that a high roll variance indicates the robot cannot find firm footholds, which decreases its speed, risks damage to its body, and also hampers the perception accuracy.

The robot has been deployed in the laboratory test track with six selected terrains denoted *flat*, *grass*, *carpet*, *cubes*, *ramp*, and *stairs* that display specific interaction properties when traversed by the hexapod walking robot as follows.

- PVC flooring (*flat*) represents an easy to traverse terrain.
- Turf-like carpet (*grass*) and red carpet (*carpet*) are soft terrains with faint response to the foot contact.

- Wooden blocks (*cubes*) with different height and slope with the base of 10×10 cm represent a harder to traverse rough terrain mock-up with the overall size of 2.3×1.2 m.
- Hard to traverse ramp (*ramp*) and wooden stairs (*stairs*) with 4 cm steps inducing vibrations due to slippage.

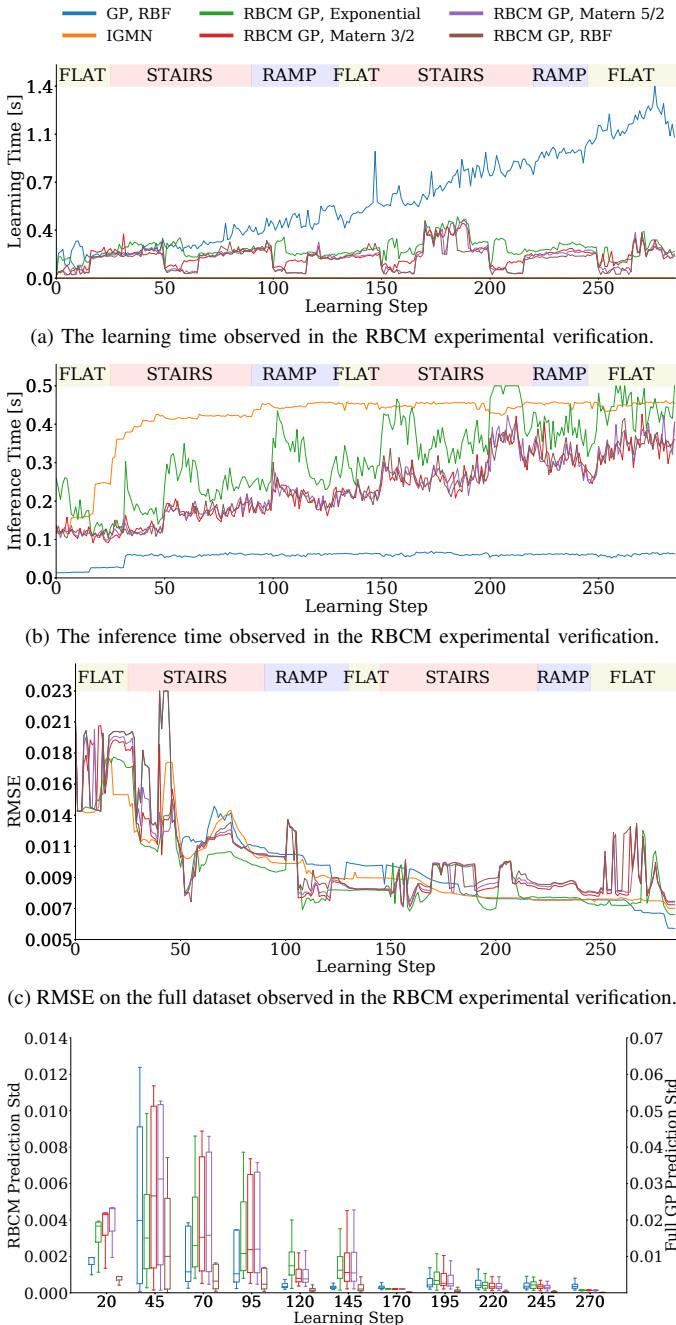
TABLE I
INDIVIDUAL PARTS OF THE DEVELOP ROBOTIC SYSTEM

Part / Module	Used Setup / Utilized Implementation
Robot	Hexapod walking robot with six legs each with three actuated joints. The robot dimensions are about 45×40 cm when standing in a default configuration, and we set $h_{max} = 0.2$ m and $\delta_{impassable} = 0.25$ m.
Locomotion control	We employ the available approach [10] with the mean walking velocity of the robot around 0.05 ms^{-1} . The robot employs the follow the carrot algorithm with 20 cm distance threshold for the path following.
Exteroception, Proprioception	Intel RealSense D435 (RGB-D imagery, 640×480 at 15 Hz), Intel RealSense T265 (localization 200 Hz).
Computational resources	Intel Core i7-8650u CPU with 16 GB RAM, Ubuntu 18.04, and implementation in ROS Melodic [24].
Environment map	\mathcal{M}_{cost} grid map with the grid cell size 10 cm, which roughly corresponds to the robot step length for a single gait cycle, and thus we chose $\delta_{desc} = 0.2$ m and $\delta_{score} = 0.5$ m to prefer larger patches of terrains.
Traversability cost	The variance of the roll angle θ_{roll} over the sliding window 10 s long.

A. Comparison of the RBCM and Pure GP-based Regression

In this experiment, we compare the performance of the herein utilized RBCM with GP regressor experts with the pure GP-based regression using real dataset collected by the used robot in the laboratory test track. We are specifically focused on the evaluation of the prediction abilities and computational requirements in the online incremental learning setup, and therefore, we consider six learning setups: the RBCM with Exponential, Matérn 3/2, Matérn 5/2, and RBF kernels; Incremental Gaussian Mixture Network (IGMN) [22], which is a representative of a broader set of fast incremental approaches; and pure GP learned using all the available observations, which serves as the baseline approach that is however expected to be computationally demanding. The individual GP experts and the pure GP model are learned using the GPy framework [12], and each of the individual experts is optimized using the limited memory Broyden-Fletcher-Goldfarb-Shanno with boundaries (L-BFGS-B) [7] limited to 200 steps.

All the methods are evaluated on the dataset containing 292 data points of terrain descriptors paired with the corresponding stability based traversal cost which ranges in $[0.005, 0.050]$, with a considerable variance over the wooden stairs. The dataset comprises two human operated runs over the laboratory test track. Each model is incrementally learned on the dataset; i.e., a single observed data point is added to the model at each learning step. Besides, at each step, the models are used to predict the traversal cost for the whole dataset, and the prediction is compared to the set of the collected measurements. Since the selected traversal cost encodes the robot experience with traversal of terrains, it cannot be obtained without the



(d) The model confidence represented as the model prediction standard deviation. Note, GP uses the alternative scale.

Fig. 5. Results of the comparison of the RBCM and Pure GP-based regression.

measurement noise caused by the robot motion, and thus the set is considered to represent the traversal cost ground truth. We also report the time to learn and predict the traversability costs of the whole dataset. We have processed the dataset in 100 learning-and-prediction trials, and report the mean value to profile the proposed system and mitigate the effects of OS scheduling, garbage collection, etc. The computational times to learn the individual models, the inference times, and to predict means and standard deviations are reported in Fig. 5.

The results indicate that the fastest learning method is the IGMN, which fuses a new observation with $\mathcal{O}(1)$, although

the theoretical time complexity of the RBCM is the same for the constant m_{\max} , as only one expert of the limited size needs to be retrained. The GP has the fusing complexity $\mathcal{O}(n^3)$ since it retrains the whole model at each time step. The inference times reported in Fig. 5b slowly increase for the RBCM as the number of experts gradually increases. The evolution of the Root Mean Squared Error (RMSE) is shown in Fig. 5c. The peaks in the RBCM correspond to the influence of the newly inserted small experts with a low number of observations, which are overly confident due to a few observations. Thus, it is desirable to set the expert minimum size m_{\min} in the inference. At the end of the testing trail, the IGMN provided similar results to most of the RBCM variants, but the GP learned the best representation of the terrain costs; however, the RBCM models outperform the GP before the second traverse of the experimental mock-up when the robot revisits already visited areas. Even though the IGMN performs similarly to the RBCM, its main drawback is that it does not predict model confidence. The results in Fig. 5d show that the predictive standard deviation of the RBCM models is considerably lower than that of the GP. Both the RBCM and GP exhibit similar behavior when the predictive standard deviation is lowered by adding observations, and the predictive standard deviation for unobserved terrains is higher than the predictive standard deviation for rough terrains with varied traversal cost measurement.

In conclusion, the RBCM with GP regressor experts provides similar performance to the GP regressor regarding both performance indicators, the RMSE, and the model confidence observed on the verification dataset. Moreover, the RBCM learning is less computationally demanding and can satisfy the real-time requirements of the online deployment with the real walking robot. On the other hand, the RBCM inference time is higher, resulting in a tradeoff between the complexity of learning and inference, with a preference on the learning speed in the herein presented deployment. The best performing RBCM with the size $m_{\max} = 50$, $m_{\min} = 25$ and the exponential kernel is utilized in the autonomous exploration deployment reported in the following section.

B. Autonomous Exploration

In the experimental deployment of the proposed system in fully autonomous exploration, the laboratory test track has been surrounded by boards and boxes to guarantee the exploration is finished within a reasonable time. The considered terrains for this experiment are the *flat* ground, the wooden *cubes*, the artificial *grass*, and red *carpet*. During the experiment, the robot has been placed in the arena and requested to explore the whole area and build the terrain traversal cost model. The verification of the system performance is made by observing the robot behavior and profiling the implementation.

During the operation, the proposed system fully exploits the computational resources. The update rate and CPU usage for each individual part of the system are reported in Table II. The majority of the resources has been consumed by the implemented exploration node that is responsible for the

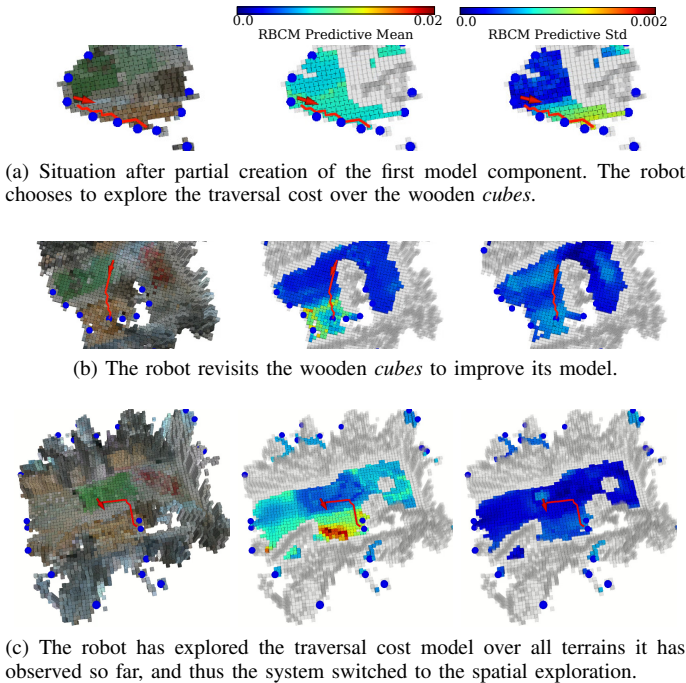


Fig. 6. Evaluation of the traversable terrain at partial time instants of the autonomous exploration. From left: the robot position on the test track, predicted traversal cost, and confidence of the traversal cost model.

construction of $\mathcal{M}_{\text{cost}}$ and exploration. At the beginning of the experimental trail, the exploration node took 1.1 s to plan a new path for a map of 219 traversable cells, in the middle of the experiment it took 10.0 s for 793 traversable cells, and at the end 23.2 s for a map with 910 cells. The speed of the $\mathcal{M}_{2.5D}$ update fluctuated based on the number of concurrently updated nodes. However, the performance of the exteroception, proprioception, model learning, and locomotion control have been stable.

TABLE II
PERFORMANCE OF THE AUTONOMOUS EXPLORATION

Process	CPU usage*	Update rate
Exteroception	36%	15 Hz
Proprioception - Traversal cost calculation	54%	200 Hz
$\mathcal{M}_{2.5D}$ and \mathcal{M}_{trv} construction and terrain descriptors calculation	98%	1–30 Hz ⁺
Model learning	61%	1 Hz
$\mathcal{M}_{\text{cost}}$ inference and exploration	450%	0.05–1 Hz
Locomotion control	19%	–

* Maximal CPU usage 800% (4 cores with Hyper-threading)

⁺ Depending on the size of the map update

The exploration deployment has been performed several times with similar behavior. The hexapod walking robot has been deployed using either the *adaptive* motion gait [10], which is designed to traverse rough terrains, or with the *regular* gait, which is faster but less capable on rough terrains. The value of λ_{score} has been set to 0.001 and 0.0005 for the *adaptive* and *regular* gait, respectively.

The behavior of the robot in the selected showcase situations is documented in Fig. 6, and the complete evolution of the spatial and traversal cost models over one trial is presented in the accompanying video. Altogether 11 experts with $m_{\text{max}} = 50$ have been created in the total during this trial. At the beginning of each experiment, the robot is located on the *flat* ground and it is allowed to freely explore the environment. As the robot did not yet traverse any terrain, the terrain traversal cost and the model confidence are both uniform, and the robot chooses to go towards the nearby spatial goal. After the first model component is created, the robot typically explores some area with a low model confidence (represented by a high predictive standard deviation), e.g., the wooden *cubes* or the centrally located artificial *grass*, see Fig. 6a. Occasionally, the robot briefly samples traversal cost over a terrain type, and continues on to explore another terrain that is unknown. In such a case, the robot may return to the first terrain to further improve its model, see Fig. 6b. If the robot has observed all the readily available terrains, it continues with spatial exploration, see Fig. 6c. Although it is not possible to provide ground truth for the learned model, because it depends on the particular trial, the robot has always identified the *flat* ground, artificial *grass*, and red *carpet* as an easy to traverse, and the rough wooden *cubes* as hard to traverse.

Based on the experimental deployment, we can conclude that the robotic system presented in this paper is capable of exploiting both the spatial and traversal cost model exploration, and selects the one that suits its currently accumulated knowledge with the preference towards traversal cost model exploration. The system is also capable of making informed decisions and intentionally avoid hard-to-traverse areas.

VI. CONCLUSION

In this paper, we present a robotic system for spatial exploration that is combined with the exploration of underlying traversal cost model over traversable terrains that is enabled by employing the Robust Bayesian Committee Machine (RBCM) with GP regressor experts, which learning part is less computationally demanding than the pure GP regressor, and thus it is more suitable for online decision-making. The additional advantage of using the RBCM (e.g., in comparison to the IGMN) to create the terrain traversal cost model incrementally is that it provides both the predictive mean and variance for the observed terrains, and thus allows the robot to explore areas of low model fidelity. The proposed approach has been deployed in the developed robotic system and verified in a fully autonomous exploration with the hexapod walking robot. The reported experimental results support the robot is capable of exploiting the spatial knowledge and make informed decisions and intentionally avoid hard-to-traverse terrains during the exploration. We aim to extend the approach by incorporating a combined probabilistic representation of the terrain traversal feasibility and terrain traversal cost, and also consider intended robot maneuvers, because the terrain traversability differs based on the particularly performed robot maneuver.

REFERENCES

- [1] R. Bajcsy. Active Perception. *Technical Reports (CIS)*, March 1988.
- [2] S. Bartoszyk, P. Kasprzak, and D. Belter. Terrain-aware motion planning for a walking robot. In *RoMoCo*, pages 29–34. IEEE, 2017. doi: 10.1109/RoMoCo.2017.8003889.
- [3] D. Belter, J. Wietrzykowski, and P. Skrzypczynski. Employing Natural Terrain Semantics in Motion Planning for a Multi-Legged Robot. *Journal of Intelligent & Robotic Systems*, pages 1–21, 2018. doi: 10.1007/s10846-018-0865-x.
- [4] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim. Contact Model Fusion for Event-Based Locomotion in Unstructured Terrains. In *ICRA*, pages 1–8, 2018. doi: 10.1109/ICRA.2018.8460904.
- [5] D. M. Bradley, J. K. Chang, D. Silver, M. Powers, H. Herman, P. Rander, and A. Stentz. Scene Understanding for a High-mobility Walking Robot. In *IROS*, pages 1144–1151. IEEE, 2015. doi: 10.1109/IROS.2015.7353514.
- [6] M. Brunner, B. Brüggemann, and D. Schulz. Rough Terrain Motion Planning for Actuated, Tracked Robots. In *ICAART*, pages 40–61. Springer, 2013. doi: 10.1007/978-3-662-44440-5_3.
- [7] R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, September 1995. doi: 10.1137/0916069.
- [8] M. Deisenroth and J. W. Ng. Distributed Gaussian Processes. *arXiv:1502.02843*, July 2015.
- [9] J. Faigl, M. Kulich, and M. Přebil. Goal Assignment using Distance Cost in Multi-Robot Exploration. In *IROS*, pages 3741–3746. IEEE, 2012. doi: 10.1109/IROS.2012.6385660.
- [10] Jan Faigl and Petr Čížek. Adaptive Locomotion Control of Hexapod Walking Robot for Traversing Rough Terrains with Position Feedback Only. *Robotics and Autonomous Systems*, 116:136–147, 2019. doi: 10.1016/j.robot.2019.03.008.
- [11] P. Giguere and G. Dudek. Clustering Sensor Data for Terrain Identification using a Windowless Algorithm. In *Robotics: Science and Systems IV*. Robotics: Science and Systems Foundation, June 2008. doi: 10.15607/RSS.2008.IV.004.
- [12] GPY. GPY: A gaussian process framework in python. <http://github.com/SheffieldML/GPY>, since 2012.
- [13] T. Homberger, M. Bjelonic, N. Kottege, and P. V. K. Borges. Terrain-dependant Control of Hexapod Robots using Vision. In *ISER*, pages 92–102. Springer, 2016. doi: 10.1007/978-3-319-50115-4_9.
- [14] M. G. Jadidi, J. V. Miro, and G. Dissanayake. Gaussian processes autonomous mapping and exploration for range-sensing mobile robots. *Autonomous Robots*, 42(2):273–290, February 2018. doi: 10.1007/s10514-017-9668-3.
- [15] N. Kottege, C. Parkinson, P. Moghadam, A. Elfes, and S. P. N. Singh. Energetics-informed hexapod gait transitions across terrains. In *ICRA*, volume 2015-June, pages 5140–5147. IEEE, 2015. doi: 10.1109/ICRA.2015.7139915. 10.1109/ICRA.2015.7139915 978-1-4799-6923-4.
- [16] M. Kragh, R. N. Jørgensen, and H. Pedersen. Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data. In *ICVS*, volume 9163, pages 188–197. Springer, 2015. doi: 10.1007/978-3-319-20904-3_18.
- [17] B. Lin and S. Song. Dynamic modeling, stability and energy efficiency of a quadrupedal walking machine. In *ICRA*, pages 657–670. IEEE, May 1993. doi: 10.1002/rob.8104.
- [18] R. B. McGhee and A. A. Frank. On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 3:331–351, August 1968. doi: 10.1016/0025-5564(68)90090-4.
- [19] W. Mou and A. Kleiner. Online Learning Terrain Classification for Adaptive Velocity Control. In *SSRR*, pages 1–7. IEEE, 2010. doi: 10.1109/SSRR.2010.5981563.
- [20] S. O’Callaghan, F. T. Ramos, and H. Durrant-Whyte. Contextual occupancy maps using Gaussian processes. In *ICRA*, pages 1054–1060. IEEE, May 2009. doi: 10.1109/ROBOT.2009.5152754.
- [21] K. Otsu, M. Ono, T. J. Fuchs, I. Baldwin, and T. Kubota. Autonomous Terrain Classification with Co- and Self-Training Approach. *RAL*, 1(2):1–6, 2016. doi: 10.1109/LRA.2016.2525040.
- [22] R. Pinto and P. Engel. A Fast Incremental Gaussian Mixture Model. *PLOS*, page e0141942, 2015. doi: 10.1371/journal.pone.0139931.
- [23] M. Prágr, P. Čížek, and J. Faigl. Cost of Transport Estimation for Legged Robot Based on Terrain Features Inference from Aerial Scan. In *IROS*, pages 1745–1750. IEEE, 2018. doi: 10.1109/IROS.2018.8593374.
- [24] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
- [25] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006.
- [26] A. V. Ruiz and C. Olariu. A general algorithm for exploration with Gaussian processes in complex, unknown environments. In *ICRA*, pages 3388–3393, Seattle, WA, USA, May 2015. IEEE. doi: 10.1109/ICRA.2015.7139667.
- [27] B. Sofman, E. Lin, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz. Improving Robot Navigation Through Self-Supervised Online Learning. *Journal of Field Robotics*, 23(11-12):1059–1075, 2006. doi: 10.1002/rob.

20169.

- [28] A. Stelzer, H. Hirschmüller, and M. Görner. Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *International Journal of Robotics Research*, 31(4):381–402, 2012. doi: 10.1177/0278364911435161.
- [29] S. B. Thrun. Exploration and model building in mobile robot domains. In *ICNN*, pages 175–180 vol.1. IEEE, March 1993. doi: 10.1109/ICNN.1993.298552.
- [30] V. Tresp. A Bayesian Committee Machine. *Neural Computation*, 12(11):2719–2741, November 2000. doi: 10.1162/089976600300014908.
- [31] V. A. Tucker. The Energetic Cost of Moving About: walking and running are extremely inefficient forms of locomotion. Much greater efficiency is achieved by birds, fish and bicyclists. *American Scientist*, 63(4):413–419, 1975.
- [32] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair. Gaussian Process modeling of large scale terrain. In *ICRA*, pages 1047–1053. IEEE, May 2009. doi: 10.1002/rob.20309.
- [33] J. Wang and B. Englot. Fast, accurate gaussian process occupancy maps via test-data octrees and nested Bayesian fusion. In *ICRA*, pages 1003–1010. IEEE, May 2016. doi: 10.1109/ICRA.2016.7487232.
- [34] P. Whaite and F. P. Ferrie. Autonomous exploration: driven by uncertainty. *Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, March 1997. doi: 10.1109/34.584097.
- [35] B. Yamauchi. A frontier-based approach for autonomous exploration. In *CIRA*, pages 146–151. IEEE, July 1997. doi: 10.1109/CIRA.1997.613851.
- [36] C. Ünsal and K. Boyer. Linearized vegetation indices based on a formal statistical framework. *Transactions on Geoscience and Remote Sensing*, 42(7):1575–1585, 2004. doi: 10.1109/TGRS.2004.826787.
- [37] P. Čížek, D. Masri, and J. Faigl. Foothold Placement Planning with a Hexapod Crawling Robot. In *IROS*, pages 4096–4101. IEEE, 2017. doi: 10.1109/IROS.2017.8206267.