Actively Improving Robot Navigation On Different Terrains Using Gaussian Process Mixture Models

Lorenzo Nardi   Cyrill Stachniss

Abstract— **Robot navigation in outdoor environments is exposed to detrimental factors such as vibrations or power consumption due to the different terrains on which the robot navigates. In this paper, we address the problem of actively improving navigation by planning paths that aim at reducing over time phenomena such as vibrations during traversal. Our approach uses a Gaussian Process (GP) mixture model and an aerial image of the environment to learn and improve continuously a place-dependent model of such phenomena from the experiences of the robot. We use this model to plan paths that trade-off the exploration of unknown promising regions and the exploitation of known areas where the impact of the detrimental factors on navigation is low, leading to an improved navigation over time. We implemented our approach and thoroughly tested it using real-world data. Our experiments suggest that our approach with no initial information leads the robot, after few runs, to follow paths along which it experiences similar vibrations or energy consumption as if it was following the optimal path computed given the ground truth information.**

I. Introduction

In outdoor environments, the choice of the specific path that a robot should follow may have a significant impact on its navigation. One of the aspects that affects navigation the most is the terrain on which the robot drives. Consider for example the scenario depicted in Fig. 1. The red path leads the robot to the goal along the shortest path. However, this might not be the best path as it follows on gravel, which causes the robot to suffer strong vibrations (see Fig. 1 right) during traversal. Strong vibrations can be problematic for the robot's perception capabilities, causing blurred camera images [26] and noisy IMU data [5] that affect localization. Furthermore, excessive vibrations might damage or affect the durability of the robot hardware. Another critical aspect for navigation is the energy consumption, particularly when deploying the robot over long time or distance [16]. Despite the longer traveled distance, the green path that in Fig. 1 follows on a paved road offers significantly lower vibrations than following the red one while requiring a similar energy expenditure. In comparison, the blue path, that follows on a paved and on a dirt road, requires less energy while offering a balanced experience in terms of traveled distance and vibrations. Therefore, it is important to take these factors into account to achieve the desired navigation behaviors.

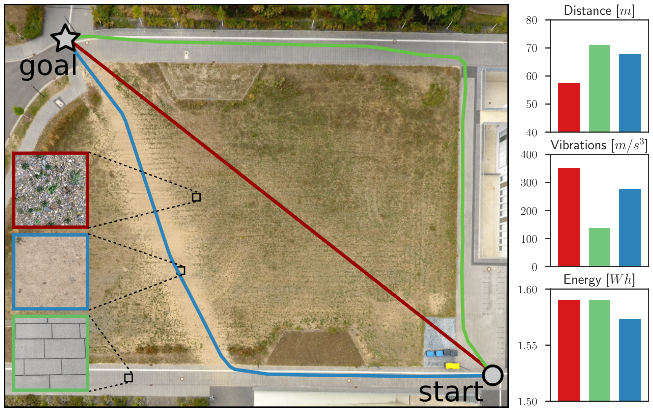

Fig. 1: Robot navigation along three paths (left) on different terrains, and the distance, the vibrations and the energy consumption that the robot experiences to navigate along them (right).

In this work, we investigate the problem of improving robot navigation by reducing detrimental factors such as high vibrations or energy consumption during traversal. One approach to achieve this is to associate a cost to these factors and to plan paths that minimize the costs. In reality, however, vibrations and energy consumption of navigating in an environment are unknown a priori, but can be observed by the robot during navigation (e.g. by an IMU or an energy monitor). There exist approaches that classify the terrain types from online perception and arbitrarily assign high costs to uneven grounds [28]. However, this might not always reflect adequately the physical quantities that the robot actually experiences while navigating. On the contrary, we aim at learning a place-dependent model from robot's observations, and at exploiting this model for path planning.

The main contribution of this paper is a novel approach to improve robot navigation over time from the experiences of the robot. We propose to use a GP mixture model [24] for learning a probabilistic model of the intensity of phenomena such as vibrations or energy consumption that affect navigation in an environment. We furthermore leverage an aerial image of the environment for modeling predictions on different terrains, without requiring training data or an explicit terrain classifier. We exploit this GP model to plan paths that aim at reducing the detrimental factors that affect navigation while taking into account the model's uncertainty in order to achieve an exploration-exploitation trade-off.

In sum, our approach is able to (i) plan paths that actively improve robot navigation in outdoor environments by reducing over time the impact of a specific phenomenon affecting navigation, (ii) learn an accurate place-dependent probabilistic model of this phenomenon on different terrains from the experiences of the robot, (iii) exploit an image prior for improving the model accuracy while considering a smaller number of observations.

All authors are with the University of Bonn, Germany.

We thank Philippe Giguere for fruitful discussions.

This work has partly been supported by the German Research Foundation under Germany's Excellence Strategy, EXC-2070 - 390732324 (PhenoRob).

II. Related Work

In outdoor navigation, the terrain may cause undesired effects on the robot such as high vibrations or power consumption. Many robot navigation system such as Obelix [9] navigate outdoors by planning paths using A^* and by trying to locally avoid rough terrains. For example, Wolf *et al.* [27] use a laser range finders to classify traversable regions based on the roughness of the terrain. Similarly, Wurm *et al.* [28] train a self-supervised classifier using the vibration of the terrain to avoid vegetated areas, while Suger *et al.* [22] classify the terrain type using a 3D-Lidar and detect obstacles accordingly. Instead, Berczi *et al.* [1] use a stereo camera to learn the terrain assessment from human demonstration. In these approaches, the cost of navigating on a specific terrain is artificially defined by the user, e.g. grass is more costly than asphalt. In contrast, we aim at learning a place-dependent model directly from the robot's observations.

Gaussian Processes [17] are commonly used to learn spatial processes from observations. For example, Lang *et al.* [10] model the 3D structure of the terrain using GPs, while DGPOM [15] maps the motion information of the dynamics in the environment. Tresp [24] introduces an approach for mixing GPs that allows for modeling different characteristics. Stachniss *et al.* [21] use this approach to model gas distribution presenting multiple modes. Similarly, we use a mixture of GPs to model the phenomena affecting robot navigation that present different characteristics on diverse terrains.

Many approaches combine GP models considering additional information. For example, Cunningham *et al.* [3] model the slippage for planetary rovers by combining GPs with a visual terrain classifier; Silver *et al.* [18] exploit satellite images to learn the traversal costs from human demonstration; while Murphy *et al.* [13] use overhead images to classify terrains and accordingly estimate the traversal costs using GP models. We incorporate information from an aerial image of the environment to learn our GP mixture model, without requiring any explicit terrain classifier.

GP models offer an attractive representation for many robotic applications. For example, Fentanes *et al.* [6] uses a GP model for exploring the soil compaction in fields. Another common application for GPs is environmental monitoring. Krause *et al.* [8] optimize the placement of the sensors by maximizing the mutual information on a GP model. Ma *et al.* [11] use a similar approach to select regions where to collect data for persistent ocean monitoring. We exploit a GP model to plan paths that aim reducing the impact of undesired effects on robot navigation while exploring unknown, promising areas.

Viseras *et al.* [25] address the problem of planning trading-off exploration-exploitation by using a sampling-based algorithm based on the mean entropy that maximizes information gathering while minimizing the path cost. Marchant and Ramos [12] instead use Bayesian optimization to plan continuous informative paths that deal with the exploration-exploitation trade off by maximizing the upper confidence bound (UCB). A similar planning approach is used by Souza *et al.* [19] for planning paths that collect data about the roughness of the terrain while minimizing the robot's vibrations. The GP-UCB [20] algorithm formalizes a utility function based on the UCB for GP optimization. Tan *et al.* [23] adopt such utility function to plan paths that adapt the exploration-exploitation trade-off for bathymetry monitoring. We use a similar concept of confidence bounds to plan paths that trade-off exploration and exploitation, but that guide the robot early towards the goal through low-cost paths, without requiring it to explore the whole environment.

III. Gaussian Process for Learning a Model for Robot Navigation

Given the robot's observations of a specific physical quantity during navigation in an environment, we aim at learning a probabilistic model of its intensity over the environment.

A. GP Regression

An effective method for modeling spatial processes from observations, such as a place-dependent cost function, is GP regression [17]. A GP for a function $f(x)$ is defined by a mean function $m(x)$ and a covariance function $k(x_i,\, x_j)$. A common choice is to set the mean function $m(x) = 0$ and to use the squared exponential covariance function:

$$k(x_i, x_j) \;=\; \varsigma_f^2 \cdot \exp\left(-\frac{1}{2}\frac{|x_i - x_j|^2}{\ell^2}\right) + \delta_{ij}\varsigma_n^2, \quad (1)$$

where $\theta = \{\ell,\, \varsigma_f^2,\, \varsigma_n^2\}$ are the so-called hyperparameters, and represent respectively the length scale ℓ, the variance of the output ς_f^2 and of the noise ς_n^2. Typically, the hyperparameters are learned from the training data by maximizing the log marginal likelihood.

Given a set of observations $\mathbf{y}$ of f for the inputs $\mathbf{X}$, GP regression allows for learning a predictive model of f at the query inputs $\mathbf{X}_*$ by assuming a joint Gaussian distribution over the samples. The predictions at $\mathbf{X}_*$ are represented by the predictive mean $\boldsymbol{\mu}_*$ and variance $\boldsymbol{\sigma}_*^2$ defined as:

$$\begin{aligned} \boldsymbol{\mu}_* &= \mathbf{K}(\mathbf{X}_*, \mathbf{X})\, {\mathbf{K}_{\mathrm{XX}}}^{-1}\, \mathbf{y}, \\ \boldsymbol{\sigma}_*^2 &= \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \,-\mathbf{K}(\mathbf{X}_*, \mathbf{X})\, {\mathbf{K}_{\mathrm{XX}}}^{-1}\, \mathbf{K}(\mathbf{X}, \mathbf{X}_*), \end{aligned} \quad (2)$$

where $\mathbf{K}_{\mathrm{XX}} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \varsigma_n^2\,\mathbf{I}$, and $\mathbf{K}(\cdot,\,\cdot)$ are matrices constructed using the covariance function $k(\cdot,\cdot)$ evaluated at the training and test inputs, $\mathbf{X}$ and $\mathbf{X}_*$.

B. GP Model for Navigation on Different Terrains

Consider for simplicity a robot that navigates in a 1D environment. It goes at first on pavement and then on gravel observing the vibrations illustrated as circles in Fig. 2. Given the observations and the corresponding locations, GP regression can learn a model of the vibrations amplitude over the environment by using Eq. (2). The resulting GP is illustrated in Fig. 2 (left) where the black dotted line is the predicted mean and the gray area represents the 2σ confidence bound of the prediction. The robot experiences vibration intensities characterized by different modes while navigating on pavement and on gravel. However, the GP tries

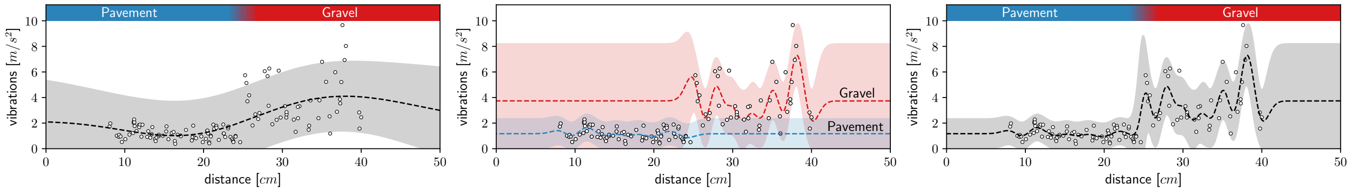

Fig. 2: Models of the vibrations during robot navigation using single GP regression (left), and using a GP mixture model (right) composed by two GPs (middle). The circles are the observed vibrations, the dotted lines are the mean prediction, the shaded areas correspond to the 2σ confidence bound of the prediction. The blue-red gradient shows the terrain on which the robot navigates.

to learn a single characteristic that fits all of the data. This results in a little informative model that smooths out the peaks and the signal trends, and that, thus, might not capture the reality well. If, for example, a peak corresponds to a stone or an hole in the ground, a smooth GP model may cause the robot to navigate through that location again.

IV. Learning a Model for Navigation on Different Terrains

Learning a GP with a single characteristic hardly fits situations in which the robot navigates on terrains with different properties. Therefore, we propose to model the intensity of a particular phenomenon that the robot experiences during traversal using a mixture of Gaussian Processes [24]. We furthermore propose to use an aerial image of the environment to learn a model that accounts for both spatial proximity and appearance similarity.

A. GP Mixture Model

A GP mixture model, first introduced by Tresp [24], is a sum of m Gaussian Processes $\{\mathrm{GP}_1, \ldots, \mathrm{GP}_m\}$ weighted according to a *gating function* that defines the probability that a data point is associated to a component of the mixture. Given the query input $\mathbf{x}_*$, the gating function $\phi_i(\mathbf{x}_*)$ represents the probability that $\mathbf{x}_*$ belongs to the model specified by GP_i. Therefore, the predictive distribution $\mathcal{N}(\mu_{\text{mix}*}, \sigma^2_{\text{mix}*})$ at $\mathbf{x}_*$ is given by

$$\begin{aligned} \mu_{\text{mix}*} &= \sum\nolimits_{i=1}^{m} \phi_i(\mathbf{x}_*)\, \mu_{i*}, \\ \sigma^2_{\text{mix}*} &= \sum\nolimits_{i=1}^{m} \phi_i(\mathbf{x}_*)\, \left(\sigma^2_{i\,*} + (\mu_{i*} - \mu_{\text{mix}*})^2\right), \end{aligned} \tag{3}$$

where $\mathcal{N}(\mu_{i*}, \sigma^2_{i\,*})$ is the predictive distribution of GP_i at $\mathbf{x}_*$. We compute the predictive mean μ_{i*} and variance $\sigma^2_{i\,*}$ for each components by using Eq. (2) and considering the probability that each data point belongs to the i-th component of the mixture. We incorporate this probability by defining the $\mathbf{K}_{\mathrm{XX}}$ term in Eq. (2) for the i-th component such that

$$\mathbf{K}_{\mathrm{XX}} \quad = \quad \mathbf{K}(\mathrm{X}, \mathrm{X}) + \varsigma_n^2\, \mathbf{\Psi}^i, \tag{4}$$

where $\mathbf{\Psi}^i$ is a diagonal matrix such that $\forall\, \mathbf{x}_j, \mathbf{\Psi}^i_{jj} = \frac{1}{\phi_i(\mathbf{x}_j)}$.

For the example introduced in Sec. III-B, we can learn a more accurate model of the vibrations amplitude by using a mixture of two GPs and a gating function defined according to the ground on which the robot navigates (see Fig. 2 upper side). Given the observations, we learn the two GPs using Eq. (2) and Eq. (4), and mix them using Eq. (3). The resulting predictive model is illustrated in Fig. 2, middle and right. As the mixture model can learn multiple characteristics, it captures the trends of the different terrains without smoothing out the peaks. It also provides better predictions in the regions where there are no observations: the mean prediction on pavement is lower than on gravel, improving the informativeness of the model.

B. Learning the Gating Function from Observations

In this work, we assume that the robot does not know on which terrain it navigates and, thus, no gating function for the mixture model is available a priori. Instead of learning an explicit terrain classifier that requires training data and a pre-defined set of classes as in [13], we compute the gating function directly from the robot's observations during navigation. To this end, first, we cluster the observations $\mathbf{y}_{\text{visited}}$ obtained at the visited locations $\mathbf{X}_{\text{visited}}$ using the *mean-shift* algorithm [7]. This is a well-know clustering algorithm which neither requires prior knowledge of the number of clusters nor constrains their shape. It computes m cluster centroids $\{C_1, \ldots C_m\}$ by iteratively shifting the mean in the direction that maximizes the density. We compute the probability that a data point $(\mathbf{x}, y)$, with $\mathbf{x} \in \mathbf{X}_{\text{visited}}$ and $y \in \mathbf{y}_{\text{visited}}$, belongs to the cluster C_i as

$$\phi_i(\mathbf{x}) \quad = \quad \frac{\exp(\|y - C_i\|)}{\sum_{j=1}^{m} \exp(\|y - C_j\|)}. \tag{5}$$

Given the probability of each observed data point to belong to each of the clusters, we can compute a gating function over the whole environment by training a classifier to predict the probability ϕ for the non-visited locations. To this end, we use m GP binary classifiers in which the inputs are the visited locations $\mathbf{X}_{\text{visited}}$ and the targets for the i-th component are the probabilities $\phi_i(\mathbf{X}_{\text{visited}})$. We consider a simple binary discriminative GP classification obtained by 'squashing' the output of GP regression into a class probability using the linear logistic function. This procedure exploits spatial proximity of the locations for learning the gating function over the whole environment.

C. Aerial Image Prior for Learning the Gating Function

Using only spatial information for learning an accurate gating function may require the robot to visit the whole environment. Even little additional information can help to speed up the learning process. One possible prior information is that terrains with similar visual appearances may similarly affect the robot navigation. We propose to use an aerial image of the environment, for example from Google Earth data, to incorporate this prior in the gating function. We consider as input to the m GP classifiers that determine the gating function both the visited locations $\mathbf{X}_{\text{visited}}$ and the corresponding colors in the aerial image $\mathcal{I}$:

$$\mathcal{I}(\mathbf{X}_{\text{visited}}) = \{r(\mathbf{X}_{\text{visited}}),\, g(\mathbf{X}_{\text{visited}}),\, b(\mathbf{X}_{\text{visited}})\}, \tag{6}$$

where r, g, b are the color channels of the image. In general, we can consider any feature that can be computed for each location in the image. As we consider different types of input to the GP classifiers, we employ an automatic-relevance-determination squared-exponential covariance function [14] that implicitly determines the 'relevance' of each dimension. Incorporating the aerial image for learning the gating function allows us for making predictions based not only on the spatial proximity but that also on the location appearances. This improves the informativeness of the model, especially in regions that have not been visited yet, reducing need to explore the whole environment.

V. Actively Improving Robot Navigation

For actively improving robot navigation, we aim at exploiting the model introduced in Sec. IV for planning paths that reduce over time the impact of a specific detrimental phenomenon. To this end, we consider a function that maps the intensity of this phenomenon to costs, and plan to minimize these costs. In this work, we define this correspondence using an identity function but, in general, any other function could be considered, also as a combination of multiple factors.

When the robot starts to navigate in a new environment, no information about the factors affecting the navigation is available and, thus, our predictive model present large uncertainty everywhere. Therefore, in the initial runs, the robot should explore the high uncertainty regions to collect informative observations for improving the model. The more accurate is the model, the more the robot should navigate through areas where the cost is likely to be low.

A. Planning to Improve Robot Navigation

We achieve this behavior by designing a tailored cost function and by computing paths that minimize it. We employ for path planning a generic graph search algorithm, such as A* or Dijkstra, that allows us to formulate planning as an explicit cost minimization problem. Therefore, we consider a grid-map representation of the environment and a corresponding discretization of our model, and we compute a path to the goal by selecting at each iteration the node that minimizes the cost of the path from the start to the node itself, often referred as g-cost.

We exploit the probabilistic nature of our predictive model to design a g-cost function that trades-off the exploration of promising non-visited regions and the exploitation of known low-cost areas. We define this g-cost at location $\mathbf{x}$, $g(\mathbf{x})$, based on the *lower confidence bound* (LCB) of the predictive distribution at $\mathbf{x}$, $\mathcal{N}(\mu,\, \sigma^2)$:

$$g(\mathbf{x}) = g(\mathbf{x}') \,+\, \|\mathbf{x}-\mathbf{x}'\| \cdot \operatorname{softplus}(\mu - \lambda\, \sigma)\,, \tag{7}$$

where $\mathbf{x}'$ is the previous location from which we expand the search, $\lambda > 0$ defines the range of the confidence bound, and the $\operatorname{softplus}$ function [4] is a rectifier.

Planning a path according to Eq. (7) minimizes at the same time the lower confidence bound of the predictions, $(\mu - \lambda\, \sigma)$, and the distance to the goal with the term $\|\mathbf{x}-\mathbf{x}'\|$. The LCB provides a natural trade-off between exploration and exploitation [2]. The term μ is small at locations with low-mean prediction, favoring low-cost paths. Whereas, the term σ represents the possible improvement of the prediction and favors paths through locations with high uncertainty that could lead to the goal through a lower-cost path. The LCB may, in general, assume negative values. In this case, minimizing Eq. (7) may result in paths that explore the whole environment rather than reaching the goal. A simple idea to overcome this issue is to use a ReLU rectifier that ensures that the values are always positive. However, using ReLU causes that the well-known low-cost locations have the same g-cost as locations with very large uncertainty. Instead, we prefer that the robot navigates through regions that are known to be low-cost over the ones that are very uncertain. Therefore, we employ as rectifier the $\operatorname{softplus}$ function that is a smooth approximation of ReLU defined as

$$\operatorname{softplus}(x) \;=\; \log(1 \,+\, \exp(x)). \tag{8}$$

Using such rectifier determines that the robot prefers low-mean and low-uncertainty areas over medium-mean high-uncertainty ones, and, in turn, prefers these areas over medium-mean low-uncertainty ones.

This planning approach leads to a navigation behavior that initially, when the predictive uncertainty σ is high, explores the environment, while minimizing the distance to the goal. This allows the robot for collecting observations at informative locations without going for long exploratory detours to reach the goal. As the robot collects observations, the uncertainty σ decreases and the mean predicted cost μ becomes more prominent in Eq. (7). This leads the robot, over time, to prefer low-cost short-distance paths over collecting new observations at unknown less-promising locations.

B. Routine to Improve Robot Navigation Over Time

Algorithm 1 Our Approach $(\mathcal{M},\, \mathcal{I})$

```
Initialize $\boldsymbol{\mu}, \boldsymbol{\sigma}^2$, and $\mathbf{X}_{\text{visited}}, \mathbf{y}_{\text{visited}}$ are empty
for each navigation task from $\mathbf{x}_{\text{start}}$ to $\mathbf{x}_{\text{goal}}$ do
    $\mathbf{X}_{\text{path}} \leftarrow$ Plan path from $\mathbf{x}_{\text{start}}$ to $\mathbf{x}_{\text{goal}}$ $(\mathcal{M}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$
    $\mathbf{y}_{\text{path}} \leftarrow$ Navigate following $\mathbf{X}_{\text{path}}$
    Update $\mathbf{X}_{\text{visited}}, \mathbf{y}_{\text{visited}}$ including $\mathbf{X}_{\text{path}}, \mathbf{y}_{\text{path}}$
    $C_{1:m} \leftarrow$ Cluster data points $(\mathbf{X}_{\text{visited}}, \mathbf{y}_{\text{visited}})$
    for each $i \in m$ do
        $\phi_i(\mathbf{X}_{\text{visited}}) \leftarrow$ Compute gating data points $(\mathbf{y}_{\text{visited}}, C_i)$
        $\phi_i(\mathbf{X}) \leftarrow$Learn gating function $(\mathbf{X}_{\text{visited}}, \phi_i(\mathbf{X}_{\text{visited}}), \mathcal{I})$
        $\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 \leftarrow$ Learn GP$_i$ $(\mathbf{X}_{\text{visited}}, \mathbf{y}_{\text{visited}}, \phi_i(\mathbf{X}))$
    $\boldsymbol{\mu}, \boldsymbol{\sigma}^2 \leftarrow$ Mix GP models $(\boldsymbol{\mu}_{1:m}, \boldsymbol{\sigma}^2_{1:m}, \phi_{1:m}(\mathbf{X}))$
```

The routine of our approach for actively improving robot navigation is illustrated in Alg. 1. It requires as input only the grid-map $\mathcal{M}$ and an aerial image $\mathcal{I}$ of the environment.

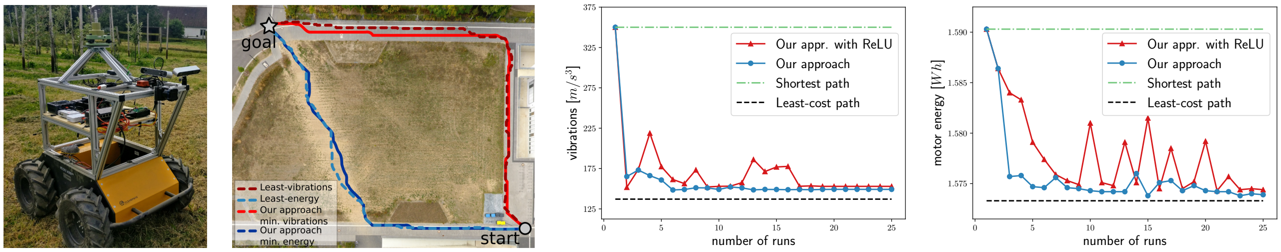

Fig. 3: Col.1: our Clearpath Husky robot. Col. 2: the least-cost paths (dotted lines) and the paths computed by our approach after 25 runs (solid lines) minimizing vibrations (red) and energy consumption (blue). Col. 3-4: the vibrations and the energy consumption experienced during robot navigation following the approaches considered in Sec. VI-B.

We initialize our probabilistic model with a mean value and high uncertainty over the whole environment (line 1). For each navigation task that the robot is required to perform, we consider the current model of the environment, denoted as $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, and plan a path using the approach described in Sec. V-A (line 3). The robot follows the path to the goal (line 4) collecting the observations $\mathbf{y}_{\text{path}}$ during navigation. Given the robot's observations, $\mathbf{y}_{\text{visited}}$, we cluster the data points using mean-shift (line 6). For each cluster centroid C_i, we compute the probability that an observation is associated to it using Eq. (5) (line 8). Using these probabilities, we compute the gating for the i-th component over the whole environment $\mathbf{X}$ (line 9), as described in Sec. IV-C. Given the gating function $\phi_i(\mathbf{X})$, we learn the corresponding model using GP regression and Eq. (4) (line 10). We combine the predictive models for each component using Eq. (3) (line 11), and so we obtain an updated model that includes the robot's observations during the last navigation task.

VI. Experimental Evaluation

The main focus of this work is an approach for actively improving robot navigation by learning a model of the phenomena affecting robot navigation and planning paths that minimize their impact over time. Our experiments are designed to show the capabilities of our method and to support our key claims, which are: (i) planning paths that reduce over time the impact of detrimental factors during navigation, (ii) learning an accurate place-dependent model for navigation on different terrains, (iii) improving the predictions while visiting a smaller number of locations by using an aerial image of the environment.

A. Experimental Setup

The experiments we present in this paper use real-world data recorded by the Clearpath Husky mobile robot illustrated in Fig. 3. In our evaluation, we consider vibrations or energy consumption as the factors which we aim at reducing to improve the navigation. We measure the intensity of the vibrations of the robot from the acceleration of an IMU along the z-axis, and the amount of energy consumption from the data provided by the motor drivers. We use an average filter to assign an intensity value to each cell of a grid-map representation of the environment. For evaluation, we drove the robot through three different environments consisting of different terrains and collected data to construct the ground truth models for vibrations and energy consumption. In our experiments, the true models are unknown to the robot, it can, however, obtain noisy observations in the locations that it visits during navigation.

B. Planning to Improve Robot Navigation Over Time

We designed the first experiment to illustrate that our approach is able to improve robot navigation by reducing over time the detrimental factors that the robot experience during traversal. To this end, we consider the environment illustrated in Fig. 1, and two scenarios. In the first one, we aim at minimizing the intensity of robot's vibrations whereas, in the second one, the amount of energy it consumes.

For each scenario, we required the robot to navigate from start to goal 25 times. Fig. 3 illustrates the vibrations (col. 3) and the energy consumption (col. 4) experienced by the robot during traversal over time. We compare our approach with the shortest path computed ignoring the vibrations/energy, and with the true least-cost paths. As no information about vibrations and energy consumption is initially available, our approach plans at first a path analogous to the shortest path. After 3-4 runs, our approach is able already to dramatically reduce vibrations and energy. After 25 iterations, the robot navigates by experiencing similar vibrations or energy consumption as it would follow the true least-cost path. As shown in Fig. 3 (col. 2), the robot follows on the paved road for reducing the vibrations (solid red), and on a paved road and dirt for reducing the energy consumption (solid blue).

We also compare our planning approach that employs a $\mathrm{softplus}$ rectifier in the g-cost function with our approach but employing a ReLU rectifier, as discussed in Sec. V-A. We show in Fig. 3 (col. 3 and 4) that, using a ReLU rectifier, the robot keeps exploring new regions rather than exploiting the known low-cost areas. Therefore, it takes longer to converge to a path along which the robot experiences low vibrations/energy consumption.

C. Learning an Accurate Model for Navigation

The second experiment aims at showing that our approach is able to learn a model that provides accurate predictions about the intensity of the phenomena affecting robot navigation on different terrains. To illustrate this, we consider the two environments depicted in Fig. 4 (col. 1), and the associated true vibrations models (col. 2). We require the robot to navigate through the locations $\mathrm{A, B, C, D}$ as shown in the figure, and to reduce over time the intensity of the vibrations that it experiences during traversal. We compare

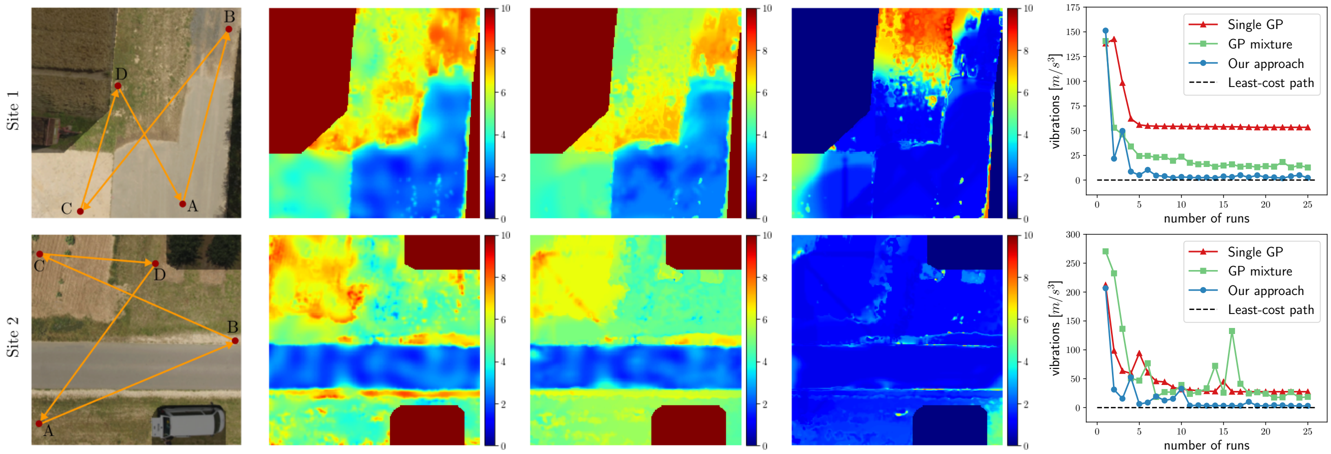

Fig. 4: Col. 1: the two environments and the navigation tasks considered for the experiments described in Sec. VI-C. Col. 2-4: the true vibrations models, the mean and the variance predictions provided by our approach after 25 runs. Col. 5: The difference of vibrations experienced by the robot following the different approaches to the least-cost path.

TABLE I: Comparison of the models considered in Sec. VI-C.

	Rms err.	Mahal. dist.	Rms err.	Mahal. dist.
Single GP	16.15	265.84	15.364	316.28
GP mixture	9.29	132.52	9.43	**150.24**
Our approach	**4.96**	**79.17**	**6.91**	151.42
	Site 1		Site 2	

our approach that incorporates an aerial image in a GP mixture model with two approaches that use the same planning algorithm but model vibrations using a single GP (*Single GP*), similar to the one described in Sec. III-B, and a mixture of GPs not considering the aerial image (*GP mixture*).

The mean and the variance of the model learned by our approach after 25 runs are illustrated in Fig. 4 (col. 3 and 4). The mean prediction shows that our approach is able to reconstruct most of the ground truth models. In particular, we obtain very accurate models for low-intensity vibrations regions as the robot prefers them for navigation. The predicted variance is mainly low, meaning that our approach is able to make reliable predictions over the whole environment. Whereas, the variance is higher at borders between different terrains where the robot observes very different vibrations in adjacent locations.

In Tab. I, we provide a comparison of the accuracy of the three models after 25 runs. We compare the root-mean-square error of the mean prediction, and the Mahalanobis distance between the predictive distributions and the true values. In Site 1, our approach presents the lowest root-mean-square error and Mahalanobis distance which is more than 3 times lower than for the single GP model. In Site 2, our approach presents a similar Mahalanobis distance as the GP mixture model, but a significantly lower error.

The accuracy of our model leads the robot to experience significantly lower vibrations already after few runs as illustrated in Fig. 4 (col. 5). After 25 runs, following our approach, the robot experiences similar vibrations as navigating along the least-cost path computed using the ground truth models (which are in practice not known to the robot). Using a single GP model, the robot stops exploring earlier and exploits a path that causes less vibrations than the shortest, but still significantly more than by following our approach. The GP mixture model shows better performance but it takes longer than our approach to find a low-cost path.

D. Advantages of Using an Aerial Image as Prior

Our approach leads quickly to an accurate model thanks to the use of an unlabeled, easy to obtain, aerial image of the environment. Using such information as prior helps to make predictions in non-visited regions. For example, few observations of low-intensity vibrations on pavement help to predict low-vibrations in areas with similar appearances and to identify promising areas for navigation. Over time, this results in a more accurate model learned by visiting a smaller number of locations. Tab. I illustrates that using an aerial image as a prior presents better or similar performance than not using it, while visiting around 20% less of the locations. Another advantage of using an aerial image is that borders between terrains are typically well defined in images. Thus, it makes easier to learn a gating function that is sharp at borders, resulting in more accurate models.

VII. Conclusion

In this paper, we presented novel approach to actively improve robot navigation on different terrains. We achieve this by using a Gaussian mixture model and an aerial image of the environment to learn a place-dependent probabilistic model of the detrimental factors that affect robot navigation, such as vibrations or energy consumption. We learn this model directly from the robot's observations without requiring any training data or explicit terrain classifier. We use this model to plan paths that deal with the exploration-exploitation trade-off, and that lead quickly the robot to navigate through low-cost paths. We implemented and evaluated our approach on different real-world environments. The experiments suggest that it is able to learn a model that provides accurate predictions about the intensity of a phenomenon in the environment, and to exploit this model to plan paths that lead the robot to navigate experiencing less vibrations or energy consumption over time.

References

[1] L. Berczi, I. Posner, and T. Barfoot. Learning to Assess Terrain from Human Demonstration Using an Introspective Gaussian-Process Classifier. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[2] D. Cox and S. John. A statistical method for global optimization. In *Proc. of the IEEE Intl. Conf. on Systems, Man, and Cybernetics (SMC)*, 1992.

[3] C. Cunningham, M. Ono, I. Nesnas, J. Yen, and W. Whittaker. Locally-Adaptive Slip Prediction for Planetary Rovers Using Gaussian Processes. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[4] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia. Incorporating second-order functional knowledge for better option pricing. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2001.

[5] J. Farrell. *GNSS Aided Navigation & Tracking: Inertially Augmented Or Autonomous*. American Literary Press Baltimore, MD, 2007.

[6] J.P. Fentanes, I. Gould, T. Duckett, S. Pearson, and G. Cielniak. 3D Soil Compaction Mapping through Kriging-based Exploration with a Mobile Robot. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3066–3072, 2018.

[7] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. on Information Theory*, 21(1):32–40, 1975.

[8] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[9] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. Autonomous Robot Navigation in Highly Populated Pedestrian Zones. *Journal of Field Robotics (JFR)*, 2014.

[10] T. Lang, C. Plagemann, and W. Burgard. Adaptive Non-Stationary Kernel Regression for Terrain Modeling. In *Proc. of Robotics: Science and Systems (RSS)*, 2007.

[11] K. Ma, L. Liu, and G. Sukhatme. Informative Planning and Online Learning with Sparse Gaussian Processes. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[12] R. Marchant and F. Ramos. Bayesian Optimisation For Informative Continuous Path Planning. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[13] E. Murphy and P. Newman. Planning most-likely paths from overhead imagery. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2010.

[14] R.M. Neal. *Bayesian Learning for Neural Networks*, volume 118. 2012.

[15] S. O'Callaghan and F. Ramos. Gaussian process occupancy maps for dynamic environments. In *Proc. of the Intl. Sym. on Experimental Robotics (ISER)*, pages 791–805, 2016.

[16] P. Ondrúška, C. Gurău, L. Marchegiani, C. Tong, and I. Posner. Scheduled perception for energy-efficient path following. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[17] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[18] D. Silver, J. Bagnell, and A. Stentz. High performance outdoor navigation from overhead data using imitation learning. *Proc. of Robotics: Science and Systems (RSS)*, 2008.

[19] J. R. Souza, R. Marchant, L. Ott, D. F. Wolf, and F. Ramos. Bayesian Optimisation for Active Perception and Smooth Navigation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[20] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. of the International Conference on Machine Learning (ICML)*, 2010.

[21] C. Stachniss, C. Plagemann, and A.J. Lilienthal. Gas Distribution Modeling using Sparse Gaussian Process Mixtures. *Autonomous Robots*, 26:187ff, 2009.

[22] B. Suger, B. Steder, and W. Burgard. Terrain-adaptive obstacle detection. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[23] Y.T. Tan, A. Kunapareddy, and M. Kobilarov. Gaussian process adaptive sampling using the cross-entropy method for environmental sensing and monitoring. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[24] V. Tresp. Mixtures of Gaussian processes. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2001.

[25] A. Viseras, D. Shutin, and L. Merino. Online Information Gathering Using Sampling-based Planners and GPs: an Information Theoretic Approach. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[26] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2012.

[27] D. Wolf, G. Sukhatme, D. Fox, and W. Burgard. Autonomous terrain mapping and classification using hidden markov models. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2005.

[28] K.M. Wurm, R. Kümmerle, C. Stachniss, and W. Burgard. Improving Robot Navigation in Structured Outdoor Environments by Identifying Vegetation from Laser Data. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009.