

Multiresolution Representations for Large-Scale Terrain with Local Gaussian Process Regression *

Xu Liu, Decai Li, Yuqing He

Abstract— To address the problem of building accurate and coherent models for large-scale terrains from incomplete and noisy sensor data, this paper proposes a novel framework that can efficiently infer terrain structures by divisionally providing the best linear unbiased estimates for the elevation values. To avoid data ambiguity caused by the uncertainty of sensor data, the proposed method introduces elevation filtering to extract the terrain surfaces, which reduces the amount of data greatly while the contained terrain information is basically unchanged. Then, for the large-scale terrains, the Gaussian mixture model is used to divide the interested regions, which remarkably improves the prediction accuracy and speed. Finally, for each subregion, a gaussian process regression model based on the static kernel is used to create a multiresolution terrain representation, which can deal with incompleteness of sensor data by considering the spatial correlations of the terrain. Evaluations of the proposed technique were conducted on diverse large-scale field terrains, including the quarry, planetary emulation terrain and highland, showing that the proposed method outperforms the state-of-art terrain modeling techniques in terms of the prediction accuracy, computation speed and memory consumption. As a practical application, the path planning problem was explored based on this terrain modeling technique to produce a better path.

I. INTRODUCTION

Building high-fidelity and efficient terrain models that can determine the traversability of quired areas is a key capacity of off-road robots in many robotics applications, ranging from autonomous navigation, exploration, and planning which are the primary needs in such scenarios like planetary exploration and field expedition. However, this remains a challenge due to some factors that mainly includes the followings: 1) Inevitable statistically dependent measurement noise that will make the data ambiguous. 2) Incomplete measurements which contains unobserved areas like partial occlusions and gaps. 3) The high computation and storage complexities caused by large-sized data in complex terrains. And these issues will be extremely intensified when robots traverse in large-scale environments with computationally constrained on-board resource. To meet the above problems, a terrain model is required to possess the following properties: 1) Probabilistic interpretation that takes the underlying uncertainty of measurements into consideration. 2) Spatial reasoning which can estimate the hidden states of

*This study was supported by the National Natural Science Foundation of China (U1608253)

Xu Liu is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016 China; Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China; University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: liuxu1@sia.cn).

Decai Li and Yuqing He is with State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: lidecai, heyuqing@sia.cn).

unobserved areas using available data. 3) Resource permitting toward enabling online navigation which requires applicable computation complexity and memory consumption.

A classical robotics mapping method that can applied to terrain representation is to directly store range measurements such as point clouds returned by laser ranger finder or stereo cameras [1]–[3]. A modified approach is to use equal-sized cubic voxels to discretize the mapping areas, in which the perceptual information is kept [4]–[6]. These methods suffer from the defects of inability to deal with data uncertainty and unbearable memory consumption increasing as the new data arriving, especially in large-scale scenarios. To mitigate these issues and generate maps which can adapt to the field terrains better, a popular terrain representation is the elevation map (EM) that uses a 2D gird-based structure to store the measured height of each cell [7]–[9]. Another method like this is mesh-based which represent the terrain using triangular or quadrilateral meshes [10], [11]. These methods are suitable for terrain multiple surface modeling, and the main drawback is the lack of considering the spatial relevance of the terrain so that they can't handle uncertainty properly and infer the terrain of unobserved areas.

To elegantly handle the above problems, as a powerful Bayesian non-parametric regression technology, Gaussian processes (GPs) [12] can be used to generate a terrain model at arbitrary resolution thorough its probabilistic reasoning ability [13]–[15], in which the terrain spatial structures can be used by covariance functions. However, these works have barely satisfactory generalization ability which may not be feasible in practical applications. GPs suffer from an $O(N^3)$ computation complexity with the number of training points, which is inapplicable for large-scale terrain, so approximation methods should be used to reduce computation complexity [16]–[18].

Facing field environments, the EM is sufficient for terrain representation and online navigation, but it lacks a statistical sound way to deal with uncertainty and probabilistic reasoning to model the unobserved areas. In this paper, mainly for terrain modeling of large-scale scenarios, a new methodology that can meet the requirements outlined above was proposed. The terrain surfaces were first extracted by elevation filtering (EF) to reduce unwanted data and raise the accuracy of regression results. Then the large mapping area is divided using Gaussian mixture model (GMM) to speed up the training of GPs which elegantly handle data uncertainty and create multiresolution models by terrain inference in a high enough precision.

The remainder of the paper is organized as follows. Section II introduces the overall framework of the proposed method and gives the process of practical application. The followed were the implementation details in Section III. The evaluation of the proposed method in a variety of large dataset and real scenario were depicted in Section IV, along with the paths planning based on the generated terrain models. Finally, section V gives the conclusion and future work.

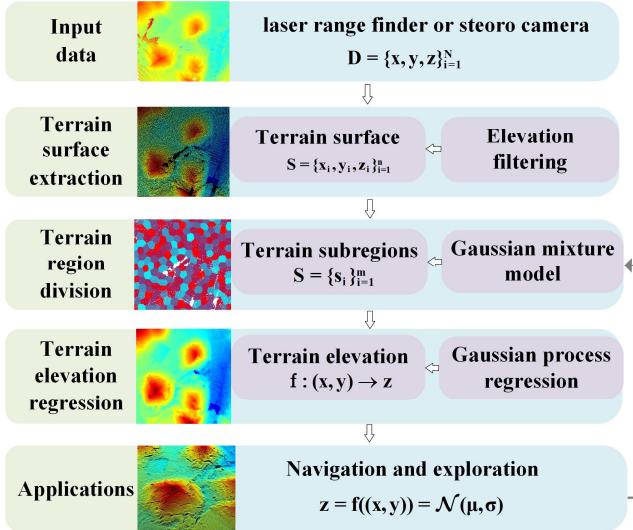


Figure 1. The overall framework of generating the terrain model for the large-scale environments.

II. FRAMEWORK AND METHODOLOGY

Before the description of the framework and methodology, the problem to be solved is defined first. Terrain measurement data are typically obtained by the end points of range sensors such as laser range finders and stereo cameras. Such data are often corrupted by uncertainty which can not only come from the sensor noise in the order of centimeters but also from the outliers arriving from dynamic objects or abnormal reflections. Meanwhile, the data may be incomplete due to the inclusion of gaps or occlusions. Particularly, for a large-scale environment, the data sampled may be too massive to be handled by the computation constrained mobile robot system. So, the actual terrain modeling problem includes the following aspects:

- 1) Building a terrain model that can fuse the uncertainty appropriately and remove the outliers effectively.
- 2) Supplementing the incomplete perception information by reasonable interpolations and generating a multiresolution terrain representation accurately.
- 3) Reducing the computation and storage complexities to enable real-time and online applications such as navigation.

Shown in Figure 1, the proposed framework can solve all the above problems together, which targets at modeling the large-scale and static environments where it can provide good enough performances (centimeter-level estimation accuracy, less memory consumption, effective confidence information and so on). Its details will be described as follows.

A. Terrain surfaces extraction

Assuming that a terrain dataset $\mathbf{D} = \{(x_i, y_i, z_i)\}_{i=1}^N$ full of noisy and missing data has been obtained from the interested regions, in which the (x_i, y_i, z_i) denotes a spatial 3D position. Typically, not all of the data is required as the initial data is too dense. Thus, the raw data needs to be sparse or sampled. Here, a method inspired from the processes of generating elevation maps was proposed to extract the surfaces of the terrain. Differently, instead of estimating elevations for each cell of a predefined 2D grid using approaches like Kalman filtering (KF) [20], the proposed method directly keeps points that have the maximal elevation among all points falling on each cell original, that is:

$$\mathbf{S} = \{(x_i, y_i, z_i) \mid z_i = \max \{z_j\}_{j=1}^{n_i}\}_{i=1}^n \quad (1)$$

where \mathbf{S} denotes the extracted terrain surface and n_i is the number of points falling into a cell. A simple formula as it is, this operation called elevation filtering (EF) plays a vital role in the framework for three reasons. Firstly, EF extracts terrain surfaces which are sufficient to represent the outdoor terrains and conduct online navigations [21], [22]. Secondly, the total data size is reduced greatly but the contained terrain spatial structures almost keep unchanged. Last and most importantly, this operation makes the data more suitable for regression analysis. Essentially, the goal of the proposed framework is to fit the relationship between the 2D coordinates (x, y) and the corresponding elevations z , but due to sensor noise and the actual terrain structures, the initial measurements data cannot satisfy one (x, y) for one z , which will result in a wrong relationship [23]. This is similar to why the classical KF is not used. For estimating the elevation of each grid cell, the result of KF is equivalent to the weighted average of elevations of points falling in this cell [24], [25], which will break the true relationship between the (x, y) and z because the estimated elevations are lower than the real ones.

By the step, a terrain surface dataset $\mathbf{S} = \{(x_i, y_i, z_i)\}_{i=1}^n$ is acquired. Notably, the outliers within the dataset have been removed previously by statistical models [26].

B. Terrain regions division

Commonly, for large-scale terrains, the data sizes are too large to be computed by limited resource even with surfaces extraction, which prevents practical applications. Lending the idea from [27] that the GP regression is a weighted sum and using the basic intuition that a point is less relevant with the points which are farther away from it, a local approximation strategy based on local cluster is proposed to divide the large mapping region and reduce the computation complexity. Here, the GMM was adopted to cluster the data and form terrain subregions because of its probabilistic results [28], which can be used to merge clusters to which a sample might belong.

A GMM comprising K components can be formulated as follows [29]:

$$P(\mathbf{x}) = \sum_{k=1}^K \alpha_k \mathcal{N}(\mathbf{x} \mid u_k, \Sigma_k) \quad (2)$$

Here \mathbf{x} is a 2D point (x, y) , $\mathcal{N}(\mathbf{x} \mid u_k, \Sigma_k)$ is a component that has mean u_k and covariance Σ_k , and α_k is its mixing coefficients. These parameters can be iteratively solved by the expectation maximization algorithm. Then the cluster to which a point \mathbf{x} belongs can be obtained by calculating the posterior probability corresponding to each component.

The underlying motivation of dividing the mapping region is that the subsequent GP regression has a cubic computation complexity of $O(n^3)$ with the number of training data, which is not practically possible to estimate the elevations using the entire data set. By decomposing computation, the method can be easily extended to large datasets. Passingly, the reason why KD-tree is not adopted is that a new set of parameters requires to be trained as a new measurement arrives, which will lose real-time property. An alternative method is to use a global parameter set, but which will cause loss of accuracy.

After this step, the whole dataset will be divided into m subsets $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^m$, where $\mathbf{s}_i = \{(x_i, y_i, z_i)\}_{i=1}^{m_i}$, $\sum_{i=1}^K m_i = m$.

C. Terrain elevations regression

For each cluster generated by GMM, which is akin to a subregion of the total mapping area, an independent GP regression (GPR) model is learnt. In nature, the training of GP model for a specific subregion is equivalent to optimizing the hyperparameters of a selected kernel from the corresponding cluster data. Therefore, the customized models can be more suitable for the local regions.

GP regression is a nonparametric approach, which define a gaussian prior probability whose mean function is generally assumed to be zero and covariance function is defined by kernel over objective function directly [12], [28], that is

$$m(\mathbf{x}) = E[f(\mathbf{x})] \equiv 0 \quad (3)$$

$$k(\mathbf{x}, \mathbf{x}') = \text{cov}(f(\mathbf{x}), f(\mathbf{x}')) \quad (4)$$

where, $m(\mathbf{x})$ is the prior mean of the objective function, $f(\mathbf{x})$ is the implicit functional relations between $f(\mathbf{x}) \triangleq z$ and $\mathbf{x} \triangleq (x, y)$, and $k(\mathbf{x}, \mathbf{x}')$ is the kernel function.

Assuming that the GP model is being learnt in subregion s_i that contains M measurements. Taking account of noise on the observed elevation z_n , then the target value is given by $t_n = z_n + \varepsilon_n$, where ε_n is a zero-mean gaussian noise with variance β . The goal of the GP regression is to find the conditional distribution $p(t_{M+1} | \mathbf{t}_M, \mathbf{X}_{M+1})$ for a new \mathbf{x}_{M+1} , where $\mathbf{t}_M = [t_1, \dots, t_M]^T$ corresponding to $\mathbf{X}_M = (\mathbf{x}_1, \dots, \mathbf{x}_M)$. To keep the notation simple, the equation will be denoted by $p(t_{M+1} | \mathbf{t}_M)$. To solve this conditional distribution, the joint distribution over \mathbf{t}_{M+1} can be first obtained by

$$\mathbf{t}_{M+1} \sim \mathcal{N}(0, \mathbf{C}_{M+1}) \quad (5)$$

where \mathbf{C}_{M+1} is the covariance matrix of the joint distribution with elements given by

$$C_{pq} = k(\mathbf{x}_p, \mathbf{x}_q) + \beta \delta_{pq} \quad (6)$$

where δ_{pq} is a Kronecker delta: $\delta_{pq} = 1$ if $p = q$ else 0. By partitioning \mathbf{C}_{M+1} , yield

$$\begin{bmatrix} \mathbf{t}_M \\ t_{M+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{C}_M & \mathbf{k} \\ \mathbf{k}^T & c \end{bmatrix}\right) \quad (7)$$

where, the scalar $c = k(\mathbf{x}_{M+1}, \mathbf{x}_{M+1}) + \beta$ and the vector \mathbf{k} has elements $k(\mathbf{x}_j, \mathbf{x}_{M+1})$ for $j = 1, \dots, M$. Here, as a Gaussian distribution, the mean $m(\mathbf{x}_{M+1})$ and variance $\text{cov}(\mathbf{x}_{M+1})$ of posterior distribution $p(t_{M+1} | \mathbf{t}_M)$ can be given by

$$m^*(\mathbf{x}_{M+1}) = \mathbf{k}^T \mathbf{C}_M^{-1} \mathbf{t}_M \quad (8)$$

$$\text{cov}(\mathbf{x}_{M+1}) = c - \mathbf{k}^T \mathbf{C}_M^{-1} \mathbf{k} \quad (9)$$

These are the key results to estimate the elevation of a query point using GP. Regarded as an interpolation technology, GPs can handle the problem of data incompleteness by inferring these results at any desired resolutions and data uncertainty by incorporating sensor noise model in the measurement data. Besides, as shown in experiment part, the probabilistic results with variances are useful to plan better paths.

In real applications, for a new query point, the subregion to which it belongs is first obtained, then the corresponding regression model is invoked to estimate its elevation value, along with the uncertainty information.

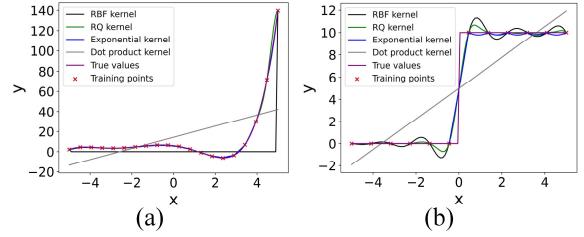


Figure 2. The comparison of adaptability of RQ covariance function and the others to abrupt changes. (a) Continuous function. (b) Discontinuous function.

III. IMPLEMENTATION DETAILS

The proposed framework entitled local Gaussian process terrain surface maps (LGPT) can meet the requirements for a terrain representation outlined in previous section, i.e., 1) extracting terrain surfaces by EF to reduce storage complexity and improve prediction precision, 2) dividing terrain region by GMM to reduce computation complexity, and 3) inferring terrain elevations by GP regression models to handle the data uncertainty and incompleteness. Thereinafter, more micro details and discussions are described to explain such macro performances.

A. Selection of cluster number

In the proposed framework, the cluster number that is the number of components of GMM is selected to divide the full terrain into subregions of corresponding number. It plays an important role in terrain division and elevation estimation. Typically, it can be selected by Bayesian information criterion (BIC) [30] given by

$$\text{BIC} = k \ln(n) - 2 \ln(L) \quad (10)$$

where k is the number of parameters to be estimated, n is the number of data points, and L is the maximized value of the likelihood function for the estimated model. Theoretically, it can select the correct number of components when there are plenty of data which are assumed to be generated from a GMM. In practice, considering the data size of each cluster which is determined by the number of components will affect the computation complexity of GP regressions, a modified selection criteria of components number yield as follows

$$\text{BIC}^* = k \ln(n) - 2 \ln(L) - \ln(k) \quad (11)$$

here, as a regularization item, the term $-\ln(k)$ tends to select models with more parameters, i.e., more components, which makes the GP regressions be realized with lower computation complexities in total due to fewer samples per cluster.

B. Selection and learning of kernel

As covariance functions in the context of GPs, the kernel functions are the crucial ingredient of GPs which define the similarity of two datapoints. Generally, even alleviated by surfaces extraction, the discontinuities result from the terrain breaks and measurement incompleteness still exist. So, the rational quadratic (RQ) covariance function [12], [31] was adopted, which is given by

$$k(x_i, x_j) = \left(1 + \frac{d(x_i, x_j)^2}{2\alpha l^2}\right)^{-\alpha} \quad (12)$$

This kernel function can be regarded as a scale mixture of squared exponential (SE) kernels with length-scale parameter $l > 0$ and scale mixture parameter $\alpha > 0$. The comparisons about adaptability of different kernels to discontinuous and abrupt changes are shown in Figure 2, in which RQ kernel is shown to possess better abilities to fit continue polynomial functions and discontinue step functions.

C. Training of the hyperparameters

The hyperparameters l and α that make the covariance functions unfixed can be learnt from the given data, using the maximum likelihood estimation. Given the terrain data set $\mathbf{X} = \{\mathbf{x}_i, y_i\}_{i=1}^M$ and $\mathbf{z} = \{f(\mathbf{x}_i)\}_{i=1}^M = \{z_i\}_{i=1}^M$, the log likelihood function of \mathbf{Z} given l , α and \mathbf{X} can be obtained by the formula for a multivariate Gaussian distribution as follows

$$\ln p(\mathbf{t} | l, \alpha) = -\frac{1}{2} \mathbf{t}^T \mathbf{C}_M^{-1} \mathbf{t} - \frac{1}{2} \ln |\mathbf{C}_M| - \frac{M}{2} \ln (2\pi) \quad (13)$$

where the hyperparameters is involved in the kernel function \mathbf{C}_M . Maximizing the likelihood $\ln p(\mathbf{t} | l, \alpha)$ is a non-convex optimization problem that can be evaluated by gradient-based methods, such as the L-BFGS-B method used here [32], [33].

IV. EXPERIMENTS

The method proposed in this paper has been evaluated by diverse terrain environments, including three natural scenarios, one synthetic scenario, and a vary large-scale terrain dataset collected by satellite. These experiments were designed to validate that the proposed method can meet the requirements outlined in Section II and can be applied to robot's navigation seamlessly. Comparisons with many other state-of-art terrains modeling techniques were conducted to show the remarkable performances of the proposed method. Finally, as a practical application, the proposed method was integrated with several popular motion planning approaches to plan a better path. All of the experiments were done on a laptop with an Intel i7 CPU and 64G RAM.

A. Evaluations and comparisons of overall performances

The first series of experiments were conducted to evaluate the terrain representation performance of the proposed method to handle uncertainty and incompleteness in the measurement data and compare it with some terrain modeling methods in terms of prediction accuracy, computational complexity and memory consumption. The used datasets are abundant which cover terrain environments of different scales, including a pit available at <http://www.pointcab-software.com/en/downloads>, a planetary emulation terrain [34], a quarry terrain collected by the drones that is from the same source as the pit dataset, and a very large-scale highland dataset gathered by the satellites. Their properties are shown in Table I.

Using these datasets, Figure 3 presents the terrain modeling results generated by the proposed framework. Figure 3(a) shows terrain surfaces extracted by the EF from the original data points collected by ranger sensors, where the unobserved areas caused by gaps and partial occlusions are clearly visible, especially in I(a), II(a), and III (a). Figure 3(b) gives the terrain maps generated using inferred elevations of random points, which demonstrates that the proposed method can deal with unobserved areas well by accurate interpolations and provide a more detailed terrain model with clearer structure. Figure 3(c)

Table I The descriptions of the used datasets.

| Terrain type | Data size | Region size (m) | Elevation range (m) |
|----------------|-----------|---------------------|-------------------------|
| pit | 23,008 | 20×21 | 444.0-448.6 |
| planetary | 50,890 | 12×13 | -2.3-1.1 |
| quarry | 301,763 | 140×135 | 5.1-21.2 |
| satellite maps | 1,326,628 | 2866×2662 | 5392.4-6146.5 |

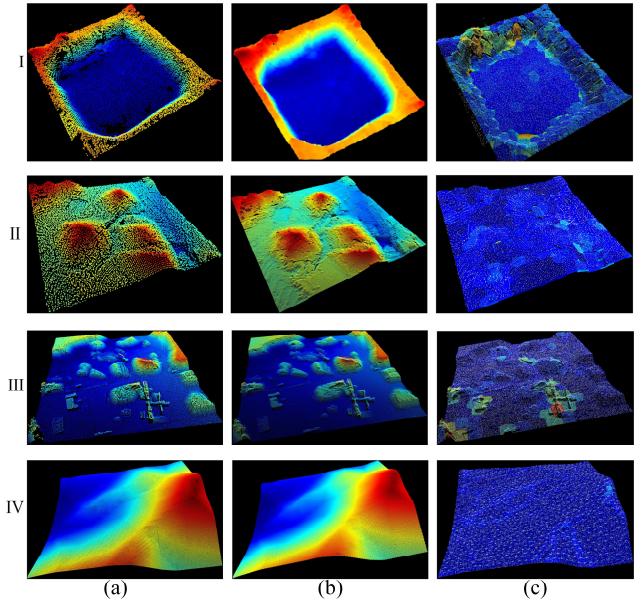


Figure 3. The terrain modeling results. Each row represents a data set: I) the pit, II) the planetary, III) the stone pit and IV) the satellite data. Each column represents a kind of terrain maps: (a) the terrain surface generated by EF, (b) the terrain inferred by GPs, and (c) the uncertainty of (b) in which color gradation represents variance of the estimates.

shows the uncertainty estimates of inference, i.e. the variances denoted by color gradations from blue to red. Notably, the points surrounded by fewer training points (shown in white) have larger variances which can be regarded as uncertainty information and applied to robot's navigation and exploration.

The comparisons of different terrain modeling techniques, including variational Hilbert regression (VHR) technique [19] and GP with non-stationary kernel functions (GPN) [15], are shown quantitatively in Table II where the prediction accuracy is denoted by the average mean square error (MSE) of ten 10-fold cross validation experiments. To be more general, the planetary emulation dataset and the quarry dataset that contain more complex terrains and discontinuous (rapidly changing) data are used. From the table results, at least two conclusions can be drawn. First, compared with VHR and GPN, the prediction accuracy of the proposed method (LGPT) is higher, which can reach centimeter level that is even smaller than the measurement noise of the range sensors. As shown later, this is mainly due to EF (elevation filtering). Second, thanks to the proposed EL, the memory consumption of LGPT is greatly reduced by at least one order of magnitude under the premise of greatly improved accuracy. Thus, the proposed method is more suitable for large-scale terrain representation. Figure 4 shows the distribution of residual errors which describes the proportions of points that less than certain residual levels in all

Table II Accuracy, speed and used memory of three terrain modeling techniques. (P: planetary terrain. Q: quarry terrain.)

| | MSE | Training time (s) | Inference time (s) | Training data size | Memory usage (Mb) |
|---|------|-------------------|--------------------|--------------------|-------------------|
| P | VHR | 0.022 | 0.975 | 1.587 | 252,616 14.65 |
| | GPN | 0.036 | 939.8 | 1.142 | 252,616 14.65 |
| | LGPT | 0.005 | 21.6 | 0.898 | 50,890 2.73 |
| Q | VHR | 0.003 | 9.266 | 8.694 | 3,571,194 72.2 |
| | GPN | 0.005 | 1151.1 | 60.743 | 3,571,194 72.2 |
| | LGPT | 0.001 | 177.1 | 4.33 | 301,763 5.82 |

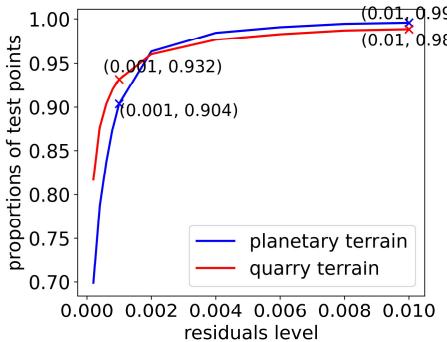


Figure 4. The proportions of points that less than certain residual error levels when using LGPT.

test points. It demonstrates that it is just a few outliers that contribute most of the residuals, while more than 99% of the residuals are less than 0.01 and 90% are less than 0.001.

The experiments above validate that the proposed method, i.e. LGPT, is capable of dealing with the uncertainties and incompleteness within the measurement data by probabilistic inferences and accurate interpolations. And particularly, for a large-scale terrain, LGPT can reduce computation and storage complexity dramatically by EF and region division.

B. Evaluations and comparisons of local performances

The second set of experiments are aimed at testifying the performances of some key components introduced in section II, which mainly focus on three aspects: 1) Reducing the data sizes and ambiguities by surfaces extraction. (2) Decreasing computation complexity by regions division. (3) Inferring terrain elevation accurately by GP regressions.

Figure 5 shows comparison of different sampling strategies applied to the data of the planetary terrain and quarry terrain, in which Figure 5(a) shows the results created by the uniform sampling (UNS), and Figure 5(b) shows those generated by the KF. The grid resolutions adopted by the two method are same to the ones of EF in Figure 3 II(a) and III(a), i.e., 0.05 m for the planetary and 0.25 m for the quarry. Commonly, the smaller the resolution, the higher the prediction accuracy and the longer the prediction time. By these comparisons, two conclusions can be drawn. First, the UNS cannot generate a terrain surface that eliminates the data ambiguity, which is required by GPRs. Second, the elevation estimated by KF is lower than the real one. This is mainly because the elevation estimated by KF is equivalent to a weighted average of all points falling into a cell. Table III gives the quantitative results about the prediction accuracy by the three sampling methods. Both the results based on UNS and KF have worse

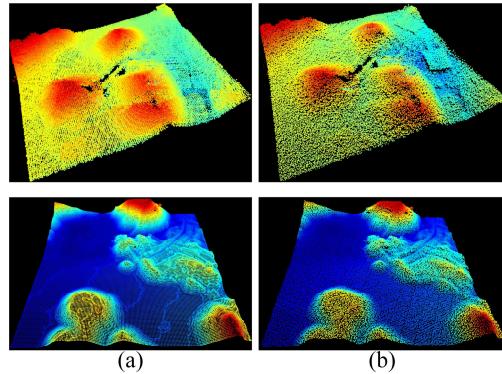


Figure 5. Outcomes of different sampling strategies applied to the original measurement data of planetary terrain (the first row) and partial quarry terrain (the second row). (a) Uniform sampling. (b) Kalman filter.

Table III Effects of different sampling methods on prediction accuracy of GPR.

| | Planetary terrain | | | Quarry terrain | | |
|----------------------|-------------------|-------|-------|----------------|-------|-------|
| | EF | KF | UNS | EF | KF | UNS |
| MSE(m ²) | 0.005 | 0.017 | 0.062 | 0.001 | 0.007 | 0.004 |

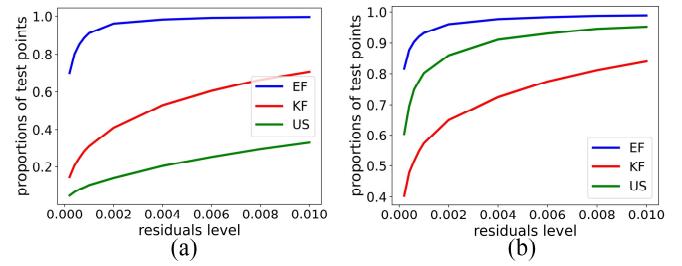


Figure 6. The final residual distribution using different sampling methods. (a) Planetary terrain. (b) Quarry terrain.

accuracy. And as mentioned in Section II, the reason is similar that the data they generate is not conducive to GPRs. This can be validated again by the residual distributions of the results as shown in Figure 6. In short, the data is supposed to be kept clean, i.e., a (x, y) corresponds to a unique z .

Figure 7 shows the results of conducting region divisions upon the planetary terrain and quarry terrain, in which the component numbers of GMM are 10 and 200, respectively. Region divisions aggregates data from adjacent regions and an individual regression model is learnt for each subregion. By this mean, the overall computation complexity of GPRs is greatly reduced. Table IV shows the quantitative results about the time consumptions of GPRs while adopting different components numbers. Normally, the computation complexity of GP regressions scales cubically with the size of data, so the consumed time is reduced greatly as the component number increases. However, the component number cannot be too big, which will make the clustering process too time-consuming. In practice, the criterion introduced in section III is adopted, that is, the component number with smaller BIC* is preferred.

The final part of this set of experiments compares GPR with some other interpolation techniques in term of prediction accuracy. The quantitative comparison results of different approaches applied to the planetary and quarry terrains are shown in Table V, including linear interpolation (LI), square

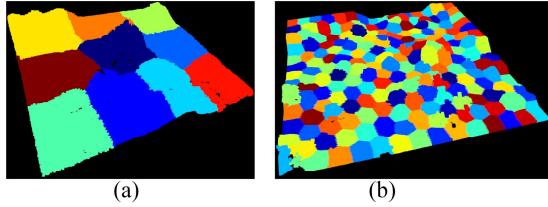


Figure 7. The results of terrain division. (a) Planetary terrain: 10 subregions. (b) Quarry terrain: 200 subregions.

Table IV The impacts of different component numbers of GMM on BIC* and GPR.

| | Planetary terrain | | | | |
|--------------------|-------------------|--------|--------|--------|--------|
| Component number | 300 | 100 | 50 | 10 | 5 |
| BIC* | 4.7e5 | 8.4e4 | 8.0e4 | 8.1e4 | 8.2e4 |
| Training time (s) | 3.83 | 2.17 | 3.27 | 27.61 | 61.47 |
| Inference time (s) | 0.66 | 0.65 | 0.79 | 2.04 | 5.35 |
| | Quarry terrain | | | | |
| Component number | 1200 | 600 | 300 | 100 | 50 |
| BIC* | 5.44e6 | 5.39e6 | 5.38e6 | 5.38e6 | 5.39e6 |
| Training time (s) | 93.7 | 132.8 | 250.4 | 1065.5 | 9363.1 |
| Inference time (s) | 4.99 | 5.32 | 6.28 | 14.43 | 302.7 |

Table V Prediction accuracy of different interpolation techniques with or without EL.

| | LI | SI | CI | KRR | GPR | |
|-----------------------|----------------|-------|-------|-------|-------|-------|
| Quarry (with EL) | 0.037 | 0.015 | 0.009 | 0.225 | 0.001 | |
| MSE (m ²) | Quarry (no EL) | 0.032 | 0.017 | 0.012 | 0.218 | 0.004 |
| Planetary (with EL) | 0.009 | 0.007 | 0.007 | 0.008 | 0.005 | |
| Planetary (no EL) | 0.055 | 0.055 | 0.055 | 0.059 | 0.058 | |

interpolation (SI), cubic interpolation (CI) and kernel ridge regression with RBF kernel (KRR). There are at least two conclusions can be drawn from the tabular results: 1) EF can effectively increase the prediction accuracy. 2) The inference accuracy of GP regression with RQ covariance function is higher than parametric polynomial fitting and KRR for terrain modeling. And considering its probabilistic results, GPR is more suitable for terrain modeling.

The above experiments validate the effectiveness of some key components constituting the proposed LGPT. It is these local properties that contribute to the overall performances demonstrated in the first set of the experiments.

C. Practical applications

As practical applications, the proposed LGPT was applied to real terrain modeling and path planning. These experiments verify that LGPT can enable robot's autonomous navigation.

The terrain modeling is oriented towards the plateau glacial environment which covers a region of roughly 521m × 298m whose altitude ranges from 5479.4 m to 5584.0 m. Figure 8(a) shows the unmanned vehicles (a UAV and a UGV) executing the terrain modeling and the red line in Figure 8(b) shows the flighthpath. The merged data from the two robots contains 34,774,307 points among which just about 306,717 are used for inference. Figure 8 (c)(d)(e)(f) show the results of LGPT, in which the final MSE of the inferred elevations are about 0.11m² that it can build the terrain model exactly.

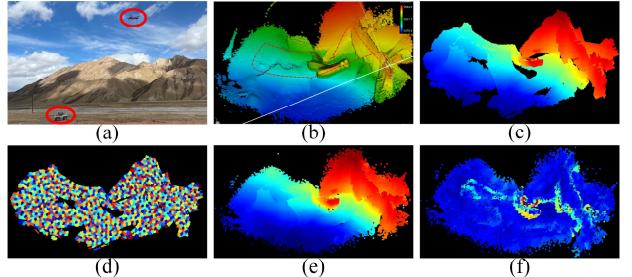


Figure 8. Terrain modeling results. (a) The plateau terrain. (b) The flighthpath of the UAV. (c) The terrain surfaces extracted by EL. (d) The result of region division. (e) The terrain inferred by LGPT. (f) The uncertainty of the inference.

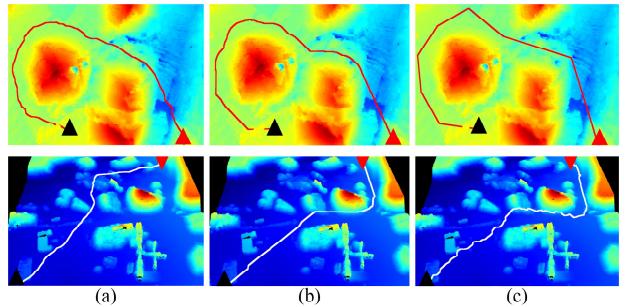


Figure 9. Path planning based on the results of terrain modeling. (a) PR. (b) A*. (c) RRT. The first and second row are the planetary terrain and quarry terrain respectively.

Finally, experiments are conducted to verify the proposed method as a terrain model based on which the path planning is implemented. Figure 9 gives the generated paths by different path planning methods, including: 1) probabilistic roadmap planner (PR) [35], 2) rapidly exploring random tree (RRT) [36], [37], and 3) A* search algorithm [38], [39]. Notably, the variance information involved in the terrain models can be used to plan better paths by avoiding areas of high uncertainty. Specifically, for the PR and RRT, the sampling process can be biased to the areas with more certainty by increasing their probability. For A* algorithm, when two grids have the same cost, the grid with less uncertainty is preferred.

V. CONCLUSION AND FUTURE WORK

In this paper, a novel terrain modeling framework was proposed toward representing the large-scale terrain scenes. Extracted from observed data, multiple terrain surfaces are clustered to partition the large-scale terrains, which enable learning an individual regression model customized for each subregion. The framework and technique are fully verified by various terrain scenarios, demonstrating that it can accurately infer terrain elevations from data containing uncertainty and incompleteness at a remarkably less computation and storage complexity. Actual terrain modeling was done in a large-scale scene with data collected by unmanned vehicles. Attributing to the variances of the posteriori estimates of elevations at different locations, better paths are planned to navigate robots in such environments. And it can also guide the robots to regions with more uncertainty in the context of exploration. In future work, the time-varying terrain textures in large-scale scenes will be considered, which can be handled as dynamic maps in the context of long-term autonomy.

REFERENCES

- [1] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, “6D SLAM—3D mapping outdoor environments,” *J. Field Robot.*, vol. 24, no. 8–9, pp. 699–722, 2007.
- [2] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, “Natural terrain classification using three-dimensional ladar data for ground robot mobility,” *J. Field Robot.*, vol. 23, no. 10, pp. 839–861, Oct. 2006, doi: 10.1002/rob.20134.
- [3] D. M. Cole and P. M. Newman, “Using laser range data for 3D SLAM in outdoor environments,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, Orlando, FL, USA, 2006, pp. 1556–1563, doi: 10.1109/ROBOT.2006.1641929.
- [4] H. Moravec, “Robot spatial perception by stereoscopic vision and 3d evidence grids,” *Perception*, 1996.
- [5] Y. Roth-Tabak and R. Jain, “Building an environment model using depth information,” *Computer*, vol. 22, no. 6, pp. 85–90, Jun. 1989, doi: 10.1109/2.30724.
- [6] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: an efficient probabilistic 3D mapping framework based on octrees,” *Auton. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013, doi: 10.1007/s10514-012-9321-0.
- [7] I.-S. Kweon, M. Hebert, E. Krotkov, and T. Kanade, “Terrain mapping for a roving planetary explorer,” in *IEEE International Conference on Robotics and Automation*, 1989, pp. 997–1002.
- [8] S. LaCroix *et al.*, “Autonomous rover navigation on unknown terrains: Functions and integration,” *Int. J. Robot. Res.*, vol. 21, no. 10–11, pp. 917–942, 2002.
- [9] R. Triebel, P. Pfaff, and W. Burgard, “Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 2006, pp. 2276–2282, doi: 10.1109/IROS.2006.282632.
- [10] I. Rekleitis, J.-L. Bedwani, D. Gingras, and E. Dupuis, “Experimental Results for Over-the-Horizon Planetary Exploration Using a LIDAR Sensor,” in *Experimental Robotics*, vol. 54, O. Khatib, V. Kumar, and G. J. Pappas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 65–77.
- [11] C. Frueh, S. Jain, and A. Zakhori, “Data Processing Algorithms for Generating Textured 3D Building Facade Meshes from Laser Scans and Camera Images,” *Int. J. Comput. Vis.*, vol. 61, no. 2, pp. 159–184, Feb. 2005, doi: 10.1023/B:VISI.0000043756.03810.dd.
- [12] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, Mass: MIT Press, 2006.
- [13] M. Ghaffari Jadidi, J. Valls Miro, and G. Dissanayake, “Gaussian processes autonomous mapping and exploration for range-sensing mobile robots,” *Auton. Robots*, vol. 42, no. 2, pp. 273–290, Feb. 2018, doi: 10.1007/s10514-017-9668-3.
- [14] V. Guizilini and F. Ramos, “Variational Hilbert Regression with Applications to Terrain Modeling,” p. 16.
- [15] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, “Gaussian process modeling of large-scale terrain,” *J. Field Robot.*, vol. 26, no. 10, pp. 812–840, 2009.
- [16] J. Quinonero-Candela, C. E. Rasmussen, and C. K. Williams, “Approximation methods for Gaussian process regression,” in *Large-scale kernel machines*, MIT Press, 2007, pp. 203–223.
- [17] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, “When Gaussian Process Meets Big Data: A Review of Scalable GPs,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–19, 2020, doi: 10.1109/TNNLS.2019.2957109.
- [18] Y. Shen, M. Seeger, and A. Y. Ng, “Fast gaussian process regression using kd-trees,” in *Advances in neural information processing systems*, 2006, pp. 1225–1232.
- [19] V. Guizilini and F. Ramos, “Variational Hilbert regression for terrain modeling and trajectory optimization,” *Int. J. Robot. Res.*, vol. 38, no. 12–13, pp. 1375–1387, Oct. 2019, doi: 10.1177/0278364919844586.
- [20] S. Thrun, “Robotic mapping: A survey,” *Explor. Artif. Intell. New Millenn.*, vol. 1, no. 1–35, p. 1, 2002.
- [21] R. Hadsell, J. A. Bagnell, D. Huber, and M. Hebert, “Accurate Rough Terrain Estimation with Space-Carving Kernels,” p. 8.
- [22] P. Pfaff, R. Triebel, C. Stachniss, P. Lamon, W. Burgard, and R. Siegwart, “Towards Mapping of Cities,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 4807–4813, doi: 10.1109/ROBOT.2007.364220.
- [23] J. O. Rawlings, S. G. Pantula, and D. A. Dickey, *Applied regression analysis: a research tool*, 2nd ed. New York: Springer, 1998.
- [24] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960, doi: 10.1115/1.3662552.
- [25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [26] V. J. Hodge and J. Austin, “A Survey of Outlier Detection Methodologies,” p. 42.
- [27] C. K. I. Williams, “Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond,” in *Learning in Graphical Models*, M. I. Jordan, Ed. Dordrecht: Springer Netherlands, 1998, pp. 599–621.
- [28] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [29] D. Reynolds, “Gaussian Mixture Models,” in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds. Boston, MA: Springer US, 2015, pp. 827–832.
- [30] G. Schwarz, “Estimating Dimension of a Model,” *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978, doi: 10.1214-aos/1176344136.
- [31] T. Hofmann, B. Schölkopf, and A. J. Smola, “Kernel methods in machine learning,” *Ann. Stat.*, pp. 1171–1220, 2008.
- [32] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A Limited Memory Algorithm for Bound Constrained Optimization,” *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, Sep. 1995, doi: 10.1137/0916069.
- [33] J. L. Morales and J. Nocedal, “Remark on ‘algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization,’” *ACM Trans. Math. Softw.*, vol. 38, no. 1, pp. 1–4, Nov. 2011, doi: 10.1145/2049662.2049669.
- [34] C. H. Tong, D. Gingras, K. Larose, T. D. Barfoot, and É. Dupuis, “The Canadian planetary emulation terrain 3D mapping dataset,” *Int. J. Robot. Res.*, vol. 32, no. 4, pp. 389–395, Apr. 2013, doi: 10.1177/0278364913478897.
- [35] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.
- [36] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” 1998.
- [37] S. M. LaValle and J. J. Kuffner Jr., “Randomized kinodynamic planning,” *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.
- [38] W. Zeng and R. L. Church, “Finding shortest paths on real road networks: the case for A*,” *Int. J. Geogr. Inf. Sci.*, vol. 23, no. 4, pp. 531–543, Apr. 2009, doi: 10.1080/13658810801949850.
- [39] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.