A Segment-Wise Gaussian Process-Based Ground Segmentation With Local Smoothness Estimation

Pouria Mehrabi and Hamid D. Taghirad
Advance Robotics & Automated Systems (ARAS)
K. N. Toosi University of Technology, Tehran, Iran.
p.mehrabi@email and taghirad@kntu.ac.ir

Abstract—In terrestrial and extraterrestrial environments, the precise and informative model of the ground and the surface ahead is crucial for navigation and obstacle avoidance. The ground surface is not always flat, the functional relationship of the surface-related features may vary in different areas of the ground, and the measurement of point clouds derived from the ground surface does not necessarily reflect smoothness. Thus, ground-related features of the ground surface must be calculated based on local or point estimates. This paper proposes a segment-wise Gaussian Process (GP)-based ground estimation method with input-dependent smoothness estimation. The value of the length scale is estimated locally for each data point resulting in a more precise estimate for the rough scenes while being not computationally complex and robust to under-segmentation, sparsity, and under-representation issues. The segment-wise task estimates a partial continuous model of the ground for each radial range segment. The simulation results show that it can give a precise continuous estimate of the ground surface in rough and bumpy scenes while being fast enough for real-time applications.

Index Terms—Gaussian Process (GP) Regression, Ground Plane Estimation, Nonstationary Kernels, Local Smoothness Estimation, Driverless Cars.

I. Introduction

In recent years, there has been an increase in demand for various types of Unmanned Ground Vehicles (UGVs). Thus, various three-dimensional perception methods have arisen to address semantic aspects of the matter [1], [2]. For this purpose, LiDAR has been extensively used due to its centimeter-level accuracy, ability to measure longer distances than cameras, and multi-directional sensing ability [3], [4]. There are three primary purposes for the captured data: tracking, detection, and semantic segmentation. ground segmentation is a semantic segmentation task. Ground segmentation aims for two primary purposes: navigation and semantic classification or the perception mode. While in the navigation mode, the algorithm finds lines navigable areas to help the UGV [5], [6], in the perception mode, it tries to precept neighboring objects to be able to track or avoid them.

In addition to improving object detection by Euclidean clustering [7], precise ground segmentation also reduces the computational cost of semantic segmentation. Although necessary for navigation and semantic classification, the ground segmentation method remains challenging for any UGV’s perception system.

Every UGV that maneuvers on a surface touch the ground. The ground model estimate for flat surfaces is obtained by filtering the point cloud using simple plane fitting algorithms such as height thresholding or RANSAC [8] or [9]. Although these methods are efficient enough for navigational objectives in flat urban scenarios, they cannot be applied where there are portions of steep slopes, bumpy sections, or rough plateaus. Furthermore, if applied for the perception, they will cause under-segmentation in the final results [10] where points related to the sloped or rough areas of the ground surface are taken into account as outliers or incorrectly merged with other objects.

Another challenge of the ground segmentation task is the sparsity of the data from the LiDAR sensor that concentrates at near-origin locations and gets sparse at longer distances to the origin. Sparsity makes it challenging to estimate the ground plane at long distances precisely. Methods tend to ignore the sparsity of the data as it does not matter much in the vicinity-of-origin area, which is insufficient for perceptional purposes as it demands perceptions of longer distances. The same issue happens for the near-origin parts of the Data where the ground plane's structure is far more complex to be estimated by a single widespread model, making the ground plane estimation methods inappropriate for representatives of the actual ground plane.

On the other hand, standard Gaussian Process GP regression cannot adapt to the variable smoothness of the data. As a result, input-dependent smoothness may be overlooked using the standard GP, while it is essential in many applications such as ground segmentation. This arises from the fact that discontinuities and sudden changes in density may occur in LiDAR’s data. Therefore, a regression method that can maintain its adaptivity to local features has merit compared to other methods. Different Methods have been proposed to estimate the structure of non-stationary covariance functions [11]. The local smoothness can be modeled by a latent process and estimated using Markov Chain Monte Carlo ap-
proaches [12], [13]. Furthermore, some methods tend to obtain mean value estimates of the kernels [14], [15].

In Gaussian process-based ground segmentation, the role of the length-scale value selection is shown to have a substantial impact on how the algorithm can predict the ground precisely [16], [17]. For the ground segmentation task [18] proposes a functional relationship of the form \( \mathcal{L} = f(r) \) to relate the length scales to the input location. This is not sufficient because, in a non-flat ground, the functional relationship will not be equitable for various regions of the ground. Furthermore, [19] assumes that length scales are obtained by performing a line extraction algorithm. However, in a non-smooth scenario, this assumption fails to describe the ground planes in different locations. Moreover, [16] tends to obtain physically motivated length-scale values by segmenting the point cloud into areas with probable common slopes using the notation of critical points.

LiDAR’s non-smooth Data is not mathematically modeled at all by the above methods. In this paper, a novel segment-wise method is proposed that estimates the continuous ground plane in a single segment using point estimates of smoothness. Smoothness is modeled as a latent stochastic process. Latent variables are predicted by their mean values. Latent variables are assumed to be Gaussian. Maximum A Posteriori (MAP) estimate is used to calculate length-scale estimates at each location. The gradient-based optimization is utilized to perform the maximization task. Modeling the input smoothness of the data enables the method to locally adjust the regression model to the ground structure in a point-wise matter. This makes the method robust to under-segmentation as the regression task is adaptive to the local behavior of the elevation data. In addition, this makes the method robust to the sparsity issue because the height estimation is based on the point-wise characteristics in sections where few points are available.

To construct the support value set \( \mathcal{D} = \{ l \in \mathcal{L}, \bar{r} \in \mathcal{R} \} \), we use line-segments which divide segments into different ranges based on local physical traits [16]. It divides the area into several segments with different structures. Thus, in the near-origin location in which more complexity is present in the data, there would be more line segments. Thus, true representatives of the segments are chosen, making the method robust to the representability issue. Furthermore, since for most UGV applications, such as driverless cars or autonomous vehicles, the behavior of the ground ahead is key, the proposed method can be used in a situation where the ground ahead of the vehicle should only be estimated quickly and precisely. The width of the area that segments include can be altered as a parameter of navigation strategy.

II. The Polar Grid Map

Data points are divided into \( M \) separate sets. Each set is then divided into a varying number of bins along the radial interval. The bin sizes are related to the radial range of the points in the corresponding segment.

![Fig. 1: A 2D View of The Constructed Grid Map.](image)

The \( P_{b_n}^m \) is the set that contains all the points in the bin number \( n \) in the \( m \)th segment. In order to reduce the further computation effort, the \( P_{b_n}^m \) set is developed to contain the needed 2D points:

\[
P_{b_n}^m = \{ p^m_i = (r_i, z_i)^T \mid p_i \in P_{b_n} \},
\]

where, \( r_i = \sqrt{x_i^2 + y_i^2} \). The ground candidate set \( P_{G_m} \) is defined as follows [18]:

\[
P_{G_m} = \{ p^m_i \mid p^m_i \in P_{b_n}^m, z_i = \min(Z_n^m) \},
\]

\( Z_n^m \) is the set of corresponding height values.

III. Nonlinear Gaussian Process Regression

IV. Problem Formulation

A. The Predictive Distribution

The nonlinear GP regression assumes a dependency of the form \( y_i = f(x_i) + \epsilon_i \) exists to explain our observations from the training set \( D \). The regression task aims to obtain the predictive ground model \( P(z^* | R^*, \mathcal{D}, \theta) \):

\[
P(z^* | R^*, \mathcal{D}, \theta) = \int P(z^*, \theta_2) \times p(\mathcal{L}^*, \mathcal{L}| R^*, \mathcal{D}, \theta) d\mathcal{L}^* d\theta_2
\]

(3)

In which \( z^* \) is the regressed value at locations \( R^* \in \mathbb{R}^{1+d} \) given the data set \( D \) and hyper-parameters \( \theta \). The parameter \( d \) is the dimension of the prediction space. The parameter \( \mathcal{R} \) is the set containing the training data. \( \theta_z \) and \( \theta_1 \) are the set containing hyper-parameters. The most probable estimation of the length-scale value is obtained:

\[
P(z^* | \mathcal{R}^*, \mathcal{D}, \theta) = P(z^* | \mathcal{R}^*, \mathcal{D}, \exp(\mathcal{L}^*), \exp(\mathcal{L}), \theta_z)
\]

(4)

where \( \mathcal{L}^* \) is the mean value estimate of the length scale. \( \mathcal{R}^* \) is the desired input locations. This predictive predicts \( z^* \) value at arbitrary locations \( r^* \) at each point.

B. Modeling The Input-Dependent Smoothness

Length scales define the width of the area where observations may affect each other. The standard GP only considers a constant length scale over the whole input space. However, in applications like ground segmentation, observations may have different length scales. For example, on a large flat plateau, height observations are strongly
related to each other, even over long distances. However, in a rough or bumpy section of the same elevation data, points may not correlate with each other or have a very small impact on each other. To tackle this problem [12] has proposed the following non-stationary covariance function, which incorporates the local characteristics at both locations to model how the data points are related.

\[
K(r_i, r_j) = \sigma_j^2 \left[ \mathcal{L}_i \right]^{\frac{1}{2}} \left[ \mathcal{L}_j \right]^{\frac{1}{2}} \left[ \frac{\mathcal{L}_i + \mathcal{L}_j}{2} \right]^{-\frac{1}{2}} \exp \left( -\frac{(r_i - r_j)^2}{(\mathcal{L}_i + \mathcal{L}_j)} \right),
\]

(5)

Even by the selection of this covariance kernel for the GP regression, the problem of selecting a credible value for the length-scales \( \mathcal{L}_i \) and \( \mathcal{L}_j \) remains unresolved [18], [20]. However, careful and credible selection of the length-scale values may lead to more reliable results even for slopped terrains [16]. While a latent GP is considered to model the length scale values, their point-wise values are obtained using mean value estimation. Equation 5 is then plugged by a mean value estimation of length scales.

C. The Case of Two Joint Gaussian Processes

Two separate GPs jointly model the ground estimation task: \( GP_z \) and \( GP_l \). The \( GP_z \) models the ground heights with \( (\mathcal{L}_i, \bar{\mathcal{L}}) \) considered to be fixed parameters. The \( GP_l \) is considered to model the length scales of the former kernel function.

The Gaussian Process for the ground height values \( GP_z \) is defined as follows:

\[
GP_z : z(r) \sim GP(z(r), k(r, r'))
\]

(6)

\( z(r) \) is the height value at the location \( r \). \( z(r) \) is the mean value estimate of the height value and \( k(r, r') \) is the covariance kernel of the form declared in Equation 5. The predictive distribution can be addressed only after that, the line segments, the support value set \( \mathcal{D} \), and consequently, the local point estimate of length-scales is obtained. The predictive ground model of Equation 4 enables estimation of height values \( z^* \) at arbitrary locations \( r^* \):

\[
\mu_{GP_z} = \bar{z} = K(r^*, r)\Gamma [K(r, r) + \sigma_z^2 I]^{-1} z,
\]

(7)

\[
\text{cov}_{z^*} = K(r^*, r_a) - K(r^*, r)\left[ K(r, r) + \sigma_z^2 I \right]^{-1} K(r_a, r)\Gamma.
\]

(8)

Furthermore, \( \bar{K}(\bar{r}, \bar{r}) \) and \( \bar{K}(\bar{r}, \bar{r}) \) are corresponding covariances for the latent process, which is assumed to be of the form below:

\[
\Rightarrow \bar{K}(\bar{r}_i, \bar{r}_j) = \sigma_f^2 \exp \left( -\frac{1}{2} \frac{(\bar{r}_i - \bar{r}_j)^2}{\sigma_l^2} \right).
\]

(9)

Thus, by defining these two kernels, the predictive distribution of the continuous ground plane is obtained. The parameter \( \mathcal{L}^* = \mu_{GP_l} \) is the point estimate of the logarithm of the length-scale at \( \bar{R}^* \).

D. Pseudo-input Selection

The training set is assumed to be equivalent to the ground candidate set \( PG_m \) to obtain the ground plane model. On the other hand, the \( \mathcal{D} = (\bar{l} \in \mathcal{L}, \bar{r} \in \mathcal{R}) \) which is the training set for the latent process must be obtained too. These locations of the test inputs for the latent kernel are chosen based on a pseudo input selection algorithm. The Physically-Motivated Line Extraction (PMLE) algorithm based on the two-dimensional line fitting method from [16] is utilized to obtain critical points in the segment. Then various line-segments are constructed [16]. Critical points are the places where the ground’s behavior changes due to different causes, such as a sudden change in slope or a sudden change in sparsity. As the critical points divide the segment into different successive bins with the same elevation behavior, the support values are chosen from these sets of bins in each line segment. To choose the support values from all the points in each line segment, the angular deviation of the points from the mean angular value of that line segment is calculated, as shown in Figure 2.

E. Learning Hyper–Parameters

Acquiring a-priori knowledge of the Hyperparameters is burdensome in many applications such as this paper. Thus, the value for \( \theta = \{ \sigma_f, \mathcal{L}, \sigma_n, \bar{\sigma}_l, \bar{\sigma}_s, \bar{\sigma}_n \} \), which are mutually independent variables, might be determined by estimation of observations \( z \). Hence, we tend to find the MAP estimates of latent length scales, which means that we take credible values that maximize the likelihood of the length scales.

Log Marginal Likelihood or the logarithm of evidence \( P(\mathcal{L} \mid \mathcal{R}, \theta) \) is the integral of likelihood times the prior. The logarithm of marginal likelihood is obtained assuming
that the prior is Gaussian \( z | R \sim \mathcal{N}(0, K) \).

\[
\log P(C|z, R, \theta) = \\
\log P(z|R, \exp(C), \theta_z) + \log P(C|R, \tilde{C}, \tilde{R}, \theta_I)
\]

\( (10) \)

F. The Single-Segment Objective Function

The objective function is valid for each segment. In each segment, the structure of the kernels is different; therefore, different hyperparameters may be assumed for the latent kernel. Therefore, the value of the objective function and its structure may vary.

\[
J(\theta) = \log P(z^m | r^m, \theta) = -\frac{1}{2} \left( (z^m)^T A_m^{-1} z^m \right) + \log \left( |A_m| \right) \\
= -\frac{1}{2} \text{tr} \left( A_m^{-1} \partial A_m \bigg|_{\theta} \right) - \frac{1}{2} \text{tr} \left( B_m^{-1} \partial B_m \bigg|_{\theta} \right)
\]

where \( A_m := K_m (r, r) + \sigma^2 I \) is the covariance of \( GP_y \) and \( B_m := K_m (\tilde{r}, \tilde{r}) + \sigma^2 I \) is the covariance matrix of \( GP \).

The Scaled Conjugate Gradient optimization method is used for the optimization task [21].

G. Gradient Calculation

The gradients of the log marginal likelihood \( L(\theta) \) are obtained using:

\[
\frac{\partial L(\theta)}{\partial \theta} = \left( (z^m)^T A_m^{-1} \partial A_m \bigg|_{\theta} A_m^{-1} z^m \right) \\
- \frac{1}{2} \text{tr} \left( A_m^{-1} \partial A_m \big|_{\theta} \right) - \frac{1}{2} \text{tr} \left( B_m^{-1} \partial B_m \big|_{\theta} \right)
\]

(12)

If the partial derivatives of \( \frac{\partial A_m}{\partial \theta} \) and \( \frac{\partial B_m}{\partial \theta} \) are obtained for all divided segments, the remaining calculations are straightforward. All the gradient calculations are detailed in [22].

H. Evaluation Parameters

Two attributes from [23] are utilized to classify each input test point. First is the normalized distance between the real height value \( z^* \) and the expected mean value of the height \( \bar{z} \) at each test location which is depicted in the second plot of Figure 3. The second attribute is the variance of the output \( z^* \). A test point will be classified as a ground point if and only if it satisfies the following inequalities at the same time:

\[
\frac{|z^* - \bar{z}|}{\sqrt{\sigma^2 n + V_z}} \leq T_d \text{ and } V \leq T_V
\]

V. Simulation

All of the simulations are performed and developed using C++ in Qt Creator. Furthermore, Point Cloud Library (PCL) is utilized for the three-dimensional data points coming from the LiDAR sensor. The (GP) library from [16] is utilized to estimate the ground plane [24]. The Optimization method is developed in C++ using the alglib library. The proposed method is applied to Data set provided by [31], which contains pre-labeled ground points. Sparse segments and rough terrains are chosen to demonstrate the efficiency of the method for under-segmentation and the sparsity issue. The results of the implementation of the algorithm are presented in Figure ??.

The proposed method is shown to give better estimation results for sparse segments. Furthermore, it is shown that the under-segmentation issue is handled by the algorithm. The Average Success Rate (ASR) evaluation parameter is calculated for 100 trials. The ASR of the proposed method is 96.57% for sloped areas, while the rate for flat areas is equal to 97.3%. The reported ASR is compared to other methods in Table II.

The proposed method shows a promising performance when compared with other methods. Although other methods fail to include non-planar or complex features of the ground, the proposed method is capable of handling sloped, rough, and bumpy situations and still be considered accurate and real-time. For instance, [7] and [10] have reported converging to a local minimum because of the big bin sizes that they assign to the point cloud [27]. On the other hand, [29], [8] and [18] fail to incorporate sloped areas. The results prove the efficiency of the proposed method to identify ground points in rough scenes, incorporate sparse data, and tackle under-segmentation while being real-time applicable, accurate, and precise.

Furthermore, In Table I, the processing time of the proposed method is compared to other available LiDAR-based ground detection methods. Each frame of the data contains approximately 70,000 points. The Velodyne LiDAR’s capture rate is about 10 fps. The reported average processing time is based on the implementation of the method on an Intel Core i7-7700HQ CPU @ 2.80GHz. The method averages operate at 46 fps, which is about 4.6 times faster than the capture rate of the LiDAR sensors. Thus the proposed method is viable for real-time applications.

Furthermore, the run-time of the proposed method is compared to the run-time of the same methods in Table I. The proposed method is not only real-time but also among the fastest methods. The results indicate that the proposed

| Method                          | Average Run-Time (ms) |
|--------------------------------|-----------------------|
| Loopy Belief [25]              | 1000                  |
| GP-Based [18]                  | 200                   |
| Fast Ground Segmentation [26]  | 94.7                  |
| Slope-robust Cascade [10]      | 76.511                |
| GP-based for Sloped Terrains   | 72.91                 |
| GPF [7]                        | 33.647                |
| R-GPD (ERASOR) [2]             | 28.328                |
| PATCHWORK [27]                 | 22.742                |
| Voxel-based [28]               | 19.31                 |
| Fast segmentation [29]         | 16.96                 |
| Enhanced Ground Segmentation   | 6.9                   |
| Joint GP Method                | 21.73                 |

TABLE I: Average Run-Time of different ground segmentation methods.
algorithm is effective on the whole Data without any data truncation, preprocessing, or data omission Figure 3. This is advantageous because it allows the detection of ground points at much higher heights or distances from the vehicle. By estimating the local smoothness, the method can adjust for the sparsity problem in LiDAR point clouds. For example, only the near-vehicle Data is taken into account in most of the ground segmentation methods because the far Data is sparse and hard to handle. On the other hand, the represent-ability issue is tackled by the introduction of the pseudo-input selection algorithm, which enables the method to base its estimation on the real physical condition of each segment.

VI. Conclusions

A segment-wise ground segmentation method with point estimates of local smoothness is proposed that is capable of performing in rough and bumpy driving scenarios. The focus of the method lies in tackling some fundamental issues of the ground segmentation task. The segment-wise representation of the point cloud is obtained by performing a radial grid mapping technique on the data. The Gaussian Process Regression, $GP$, is adapted to estimate the ground height values at each test location. It is mentioned that the Gaussian process regression ($GP$) method fails to incorporate local smoothness into account even by using a non-stationary covariance. A mean value estimation strategy for length-scale estimation is utilized to make the $GP$-based method adaptive to local attributes of the elevation data. The mean prediction strategy is adapted by introducing a second Gaussian process $GP_l$ on the length-scale values. Moreover, the pseudo-input selection method is introduced to tackle the represent-ability problem, while the local smoothness estimation enables the method to solve the sparsity issue of the 3D point clouds obtained from the LiDAR sensor. It is shown that the proposed method can be trusted in rough and bumpy scenes by solving the under-segmentation issue by being locally adaptive. The simulation results were performed on the labeled data set by [31]. The efficacy of the proposed method is proven, which outperforms other common methods both for rough and urban scenes. Determination of the ground points is performed with no data truncation, both for near-origin and far-sparse data.

TABLE II: Average Success Rate (ASR) Comparison

| Method                        | Average SR(flat/sloped) |
|-------------------------------|-------------------------|
| Fast Ground Segmentation [26] | 94.61/91.25             |
| Loopy Belief [25]             | 97.19/NR                |
| Voxel-based [28]              | NR                      |
| Enhanced Ground Segmentation [30] | 94.71/91.70          |
| Fast segmentation [29]        | NR                      |
| GP-Based [18]                 | 97.67 / NR              |
| GP-based for Sloped Terrains [16] | 96.7/93.5          |
| Joint GP Method               | 97.3/96.57              |

---

Fig. 3: Estimation result of the proposed method reported with the estimation result of two other methods. The red dots are the mean value estimation $\bar{z}$ of the ground points in each test location. The Evaluation parameter of each method is reported too. It is depicted that the evaluation parameter of the proposed method is at the minimum in comparison with the other two methods, even in the sparse sections of the data $r \in (20m, 33m)$ and even at the farthest locations.
References

[1] J. Behley, M. Garbade, A. Milloto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9297–9307.

[2] H. Lim, S. Hwang, and H. Myung, “Eraser: EGocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2272–2279, 2021.

[3] G.-M. K. S.-W. S. S.-W. Shin, Myung-Ok, “Real-Time and Accurate Segmentation of 3-D Point Clouds Based on Gaussian Process Regression,” IEEE Trans. Intell. Transp. Syst., pp. 1–15, 2017.

[4] K. Anders, M. Hämmerle, G. Miernick, T. Drewes, A. Escalona, C. Townsend, and B. Hölle, “3D GEOLOGICAL OUTCROP CHARACTERIZATION: AUTOMATIC DETECTION OF 3D PLANES (AZIMUTH AND DIP) USING LiDAR POINT CLOUDS,” ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci., vol. III-5, no. July, pp. 105–112, 2016. [Online]. Available: http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/III-5/105/2016/isprs-annals-III-5-105-2016.pdf

[5] K. Na, B. Park, and B. Soo, “Drivable space expansion from the ground base for complex structured roads,” in 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2016, pp. 000373–000378.

[6] J. Byun, K.-i. Na, B.-s. Seo, and M. Roh, Drivable Road Detection with 3D Point Clouds Based on the MRF for Intelligent Vehicle. Cham: Springer International Publishing, 2015, pp. 49–60.

[7] D. Zermas, I. Izzat, and N. Papanikolopoulos, “Fast Segmentation of 3D Point Clouds: A Paradigm on LiDAR Data for Autonomous Vehicle Applications,” IEEE Int. Conf. Robot. Autom., no. May, pp. 5067–5073, 2017.

[8] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Commun. ACM, vol. 24, no. 6, p. 381–395, Jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[9] A. Asvadi, C. Premeida, P. Peixoto, and U. Nunes, “3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes,” Rob. Auton. Syst., 2016.

[10] P. Morton, and A. Frenkel, “On the segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion,” IEEE Intell. Veh. Symp., pp. 560–565, 2010.

[11] A. Brooks, A. Makarenko, and B. Upcroft, “Gaussian process models for sensor-centric robot localisation,” in Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., 2006, pp. 56–61.

[12] C. J. Paciorek and M. J. Schervish, “Nonstationary Covariance Functions for Gaussian Process Regression,” pp. 1–27, 2003.

[13] C. J. Paciorek, “Nonstationary Gaussian processes for regression and spatial modelling,” Thesis, vol. 6, no. May, pp. 1–34, 2003.

[14] C. Plagemann, K. Kersting, and W. Burgard, “Non-stationary Gaussian process regression using point estimates of local smoothness,” Lecture Notes in Computer Science, vol. 5212, no. 2, pp. 204–219, 2008.

[15] Y. Zhang, S. Ghosh, I. Asher, Y. Ling, and L. Wang, Learning Uncertainty using Clustering and Local Gaussian Process Regression. [Online]. Available: https://arc.aiia.org/doi/abs/10.2514/6.2019-1730

[16] M. Mehrabi and H. D. Taghirad, “A gaussian process-based ground segmentation for sloped terrains,” in 2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM). IEEE, 2021, pp. 371–377.

[17] K. Liu, W. Wang, R. Thamarasa, J. Wang, and Y. Zuo, “Ground surface filtering of 3D point clouds based on hybrid regression technique,” IEEE Access, vol. 7, pp. 23270–23284, 2019.