BumpyPatch: Heightmap-based Outdoor Point Cloud Segmentation to Find Less Bumpy Road

Jiwon Park
Department of Electronic Engineering
Kyung Hee University
Yongin, Republic of Korea
overflow21@khu.ac.kr

Hyoseok Hwang
Department of Software Convergence
Kyung Hee University
Yongin, Republic of Korea
hyoseok@khu.ac.kr

***Abstract*—Autonomous mobile robots operate in a range of environments, from controlled indoor settings to unpredictable outdoor terrains. These varied conditions present challenges that require advanced navigation systems for their safe and efficient operation. A key component of this navigation is accurately assessing the ground texture. Any misjudgments can jeopardize both the robot's sensitive equipment and its carried cargo. In this research, we propose a novel method that uses heightmaps created by mapping the z-coordinates of 3D LiDAR-derived point cloud data to grayscale pixel values for evaluating outdoor ground textures. This approach effectively converts point cloud data, providing information to assist mobile robots in navigating less bumpy roads in outdoor settings. We present classification techniques for terrains based on the environment's nature: static, which pertains to individual point cloud files representing completed scenes, and dynamic, related to the real-time point cloud data captured by moving robots. For both static and dynamic environments, we introduce tailored heightmap classifiers, incorporating Inertial Measurement Unit (IMU) insights to consider the robot motion influenced by terrain texture. Our proposed method demonstrates superior performance compared to existing methods that analyze the point cloud directly and perform texture analysis with high accuracy in both static and dynamic environments. The code can be downloaded from https://github.com/zzziito/BumpyPatch.**

***Index Terms*—mobile robot, outdoor, point cloud, heightmap, autonomous driving**

I. Introduction

In recent years, autonomous mobile robots have been progressively used not just in controlled indoor settings, but also in the unpredictable and unstructured terrains of outdoor environments [1]. Their increasing presence in outdoor terrains is due to their potential in performing tasks that might be hazardous for humans, improving operational efficiency, and offering contributions in areas such as agriculture, environmental monitoring, and search and rescue missions [2]. Outdoor mobile robots are now pivotal in the delivery sector, enabling timely, contactless delivery of goods while reducing human labor and streamlining logistics.

Unlike reletively uniform indoor spaces, outdoor terrains present challenges like uneven surfaces and diverse ground textures [3]. These terrains vary widely, from unpaved surfaces like grass and mud to paved roads made of materials like concrete and brick. When traveling on paved roads, if a mobile robot cannot detect surface irregularities, it may disrupt stable navigation and damage the carried cargo.

An adaptive control strategy is crucial to ensure vehicle stability in these environments [4]. A key challenge within the field of adaptive outdoor mobile robot navigation, is the requirement for subtle ground texture assessment [5]. Outdoor landscapes showcase varied characteristics, making a binary classification like "traversable" or "non-traversable" insufficient [6]. Research by Waibel et al. suggests that these terrains demand a more nuanced approach [7]. Given that mobile robots are often tasked with transporting sensitive equipment or handling deliveries, the precise assessment of ground texture becomes vital for ensuring operational safety and efficiency [8]. By precisely assessing ground texture, robots can make well-informed navigation choices, reducing potential damage and maintaining the most effective routes.

It is essential to differentiate between ground texture assessment and semantic segmentation [9], which are standard practices in autonomous mobile robotics. While the latter methodologies divide an image or point cloud into distinct sections based on semantic properties, assessing texture involves evaluating the physical characteristics, like roughness, which are vital for real-time navigation decisions. Various sensors, such as cameras, have been explored to acquire information for autonomous mobile robot navigation [10]. Vision-based methods, though promising, often fail to discern precise terrain geometries, complicating the identification of optimal paths. Given this gap, there is a growing need for a more geometrically accurate representation that captures the topological nuances of terrains for safer navigation.

In this study, we introduce a novel approach for assessing outdoor terrain textures, enhancing the navigation of outdoor mobile robots. Our methodology utilizes grayscale heightmaps derived from 3D LiDAR-acquired point cloud data. These heightmaps, where brighter shades signify higher elevations and darker shades denote lower areas, not only streamline data processing but also facilitate the integration of established image classifiers well-known for texture classification. This accelerates decision-making across diverse terrains and is compatible with the Gazebo simulator, ensuring a smooth transition to real-world applications.

Understanding the limitations of solely relying on geometri-

979-8-3503-9574-7/23/$31.00 ©2023 IEEE
DOI 10.1109/IRC59093.2023.00050

cal point cloud analysis, we incorporated an Inertial Measurement Unit (IMU) within the Gazebo simulator. This integration led to the development of a new metric, linking road roughness with IMU data, for a comprehensive understanding of how terrains affect robot motion.

The main contributions are centered around our utilization of heightmaps derived from 3D LiDAR-acquired point cloud data and IMU to accurately delineate the texture of outdoor terrains. This method can significantly aid in enhancing navigation precision for outdoor autonomous mobile robots. Our approach not only simplifies the data processing pipeline but also has proven to yield superior results compared to traditional geometric analysis using direct point cloud in both the Gazebo simulator and real-world datasets.

II. Related Works

A. Traversability Assessment

Before delving into the nuances of texture analysis, it is paramount to understand that traversability assessment serves as a foundational cornerstone for autonomous systems in unstructured environments. While 2D LiDAR is effective for indoor settings, outdoor terrains, with their non-flat grounds and non-vertical obstacles, demand advanced techniques. Therefore, advanced sensor setups that offer a detailed 3D model of the environment, such as stereo camera, RADAR, and 3D LiDAR, or a fusion of different sensors, become crucial [11]. Traversability varies depending on the robot's specific capabilities and context, like ground clearance and stability. Especially in applications like planetary rovers, assessing traversability is vital to avoid obstacles that could tip them over [12]. However, using 3D LiDAR, while comprehensive, presents challenges, including the unstructured nature of its point cloud and high computational costs [13]. The studies by Oh et al. [14], and Lim et al. [15] suggested segmenting the ground plane in point cloud processing to reduce these costs. Moreover, terrain traversability can sometimes be assessed more efficiently using texture information from 2D images taken by monocular cameras, which undergo multiscale analysis for terrain non-uniformity detection [16].

B. Roughness Evaluation

Traditional methods for surface roughness inspection have largely revolved around the use of stylus-type instruments. However, these tools present significant limitations, especially when applied to tasks such as the one outlined in our study [17]. The inherent need for direct physical contact with these devices impedes measurement speed and constrains access to specific surfaces. The measurements from stylus readings are based on limited line samplings. This limitation can lead to inaccuracies in evaluating surface quality. For high-speed automated inspections, like our proposed task, stylus-based methods may not be the best fit. Deep-learning techniques, on the other hand, have seen significant advancements in terrain roughness evaluation, particularly in autonomous navigation. Waibel et al. [7] introduced a strategy integrating both local and global planning techniques to create continuous cost maps. Narksri et al. [18] launched a self-supervised machine learning technique tailored to gauge terrain roughness utilizing laser range data. The dataset highlighted two particularly effective roughness metrics derived from IMU z-axis acceleration. Deep-learning mechanisms come with their own set of challenges. When actual driving data, like IMU measurements, are employed as labels, they could inadvertently infuse considerable noise into the system. This noise might compromise the accuracy of the results. Therefore, meticulous consideration is needed when using and interpreting deep-learning methods for roughness evaluation.

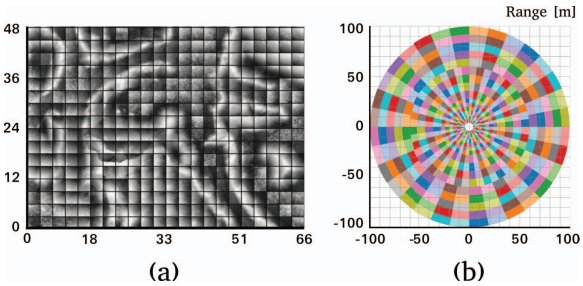

Fig. 1. Variations in cropping methods based on operational scenarios. (a) Square grid based patch of static scene. (b) Polar grid based patch of dynamic scene.

C. Texture Analysis in Grayscale Image

Texture analysis in grayscale images offers numerous techniques tailored for intricate pattern extraction. Broadly, these techniques fall into categories like statistical, model-based, and transform-based methods [19].

Statistical approaches study pixel relationships to characterize texture patterns. For instance, Local Binary Patterns (LBP) [20] interpret textures based on relative luminosity variations among neighboring pixels. Model-based strategies include fractal modeling, emphasizing texture self-similarity, and Markov Random Field (MRF) [21] that utilizes probabilistic models for inter-pixel connections. Among transform methods, Gabor filters stand out for their efficacy in analyzing textures with distinct directional and frequency orientations, as evidenced in various studies such as [22] and [23]. Deep learning, notably Convolutional Neural Networks (CNNs), further refines texture analysis by autonomously identifying intricate patterns [24].

Given the focus on images with pronounced directional stripes in this study, Gabor filters are especially relevant, highlighting their innate capability in discerning particular directional nuances. The choice of the technique often rests on the unique characteristics of the analyzed images, underscoring the need for a method-image alignment to achieve accurate texture analysis.

III. Methodology

A. Framework Overview

In our approach to analyze point cloud data, we distinctly categorized the scenes into two types: static and dynamic. This

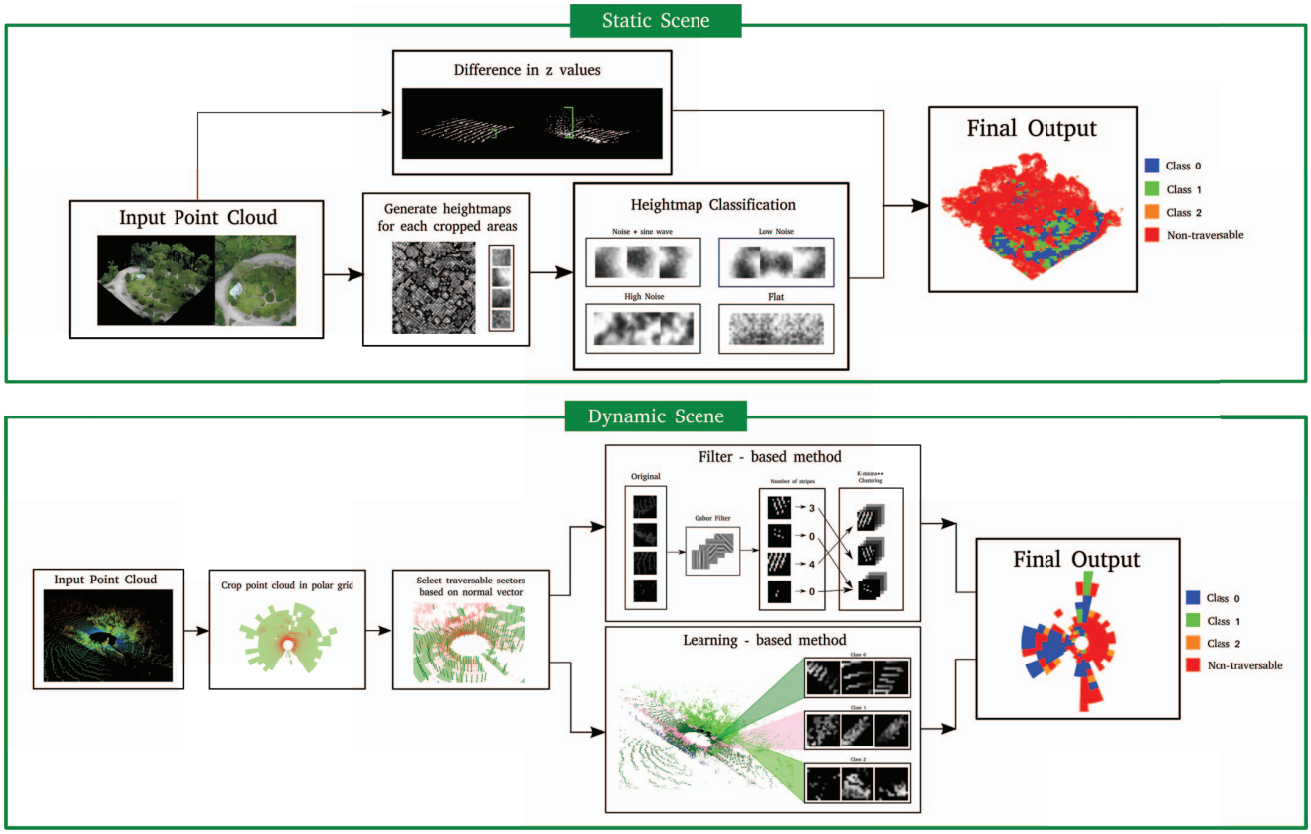

Fig. 2. Overview of BumpyPatch. BumpyPatch distinguishes point clouds into static scenes and dynamic scenes, applying different methods to each to segment rough and flat roads.

classification stemmed from the prominent issue of sparsity in point cloud data, especially evident in real-time ground texture assessment [25]. As the distance from the sensor extends, the point cloud tends to become sparser, hindering precise ground plane identification. Consequently, the demarcation between these scene types was crucial, given the inherent structural differences and the requisite unique methods of analysis.

A static scene describes a point cloud where the entirety of the space is consistently populated with points, symbolizing a complete dataset. Such scenes predominantly emerge from thoroughly scanned environments, leading to a uniformly dense file. In contrast, a dynamic scene relates to point cloud segments captured at specific time intervals, akin to those retrieved from rosbag files. These intermittent snapshots often portray varying densities, not encompassing the complete scene.

Understanding the unique challenges posed by static and dynamic scenes, we developed methodologies suited to each. Due to the differences in scene types, we used varied segmentation strategies, resulting in heightmaps of distinct configurations. These variations then guided our choice of texture classification methods. The overall system diagram can be seen in Figure 2.

B. Static Scene Analysis

In this section, we outline the pipeline, starting from the extraction of heightmaps to the evaluation of roughness in a static scene.

1) Heightmap Generation: To analyze the point cloud on a patch-by-patch basis, we divided the point cloud into $3m \times 3m$ patches. Each patch was subsequently transformed into a grayscale heightmap, sized at $15px \times 15px$ (5.13 pixels per 1 m).

2) Heightmap Patch Classification: Roughness, in the context of terrains, refers to the minute variations in the height of a surface. The mathematical representation of roughness is determined by the distribution of points. We measure roughness by observing how points are distributed on a surface, either scattered randomly or arranged in straight lines or flat patterns. It is these variations that can be emulated using patterns like noise or sine waves [26]. Recognizing the essence of terrain roughness, we sought to replicate it using a blend of techniques such as Perlin noise and sine wave combined with Gaussian blur. Small-scale undulations were represented using Perlin noise, while larger obstacles were depicted through suitable modulations of the sine wave. This careful combination enabled us to closely replicate the intricacies of real terrains. By analyzing the generated heightmaps, we identified four

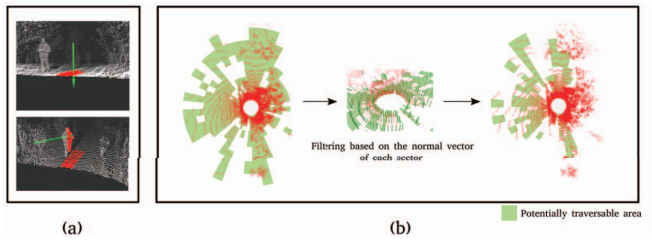

Fig. 3. Normal vector estimation and traversability analysis. (a) Estimate normal vector per patch using PCA. (b) Select potentially traversable area based on normal vector.

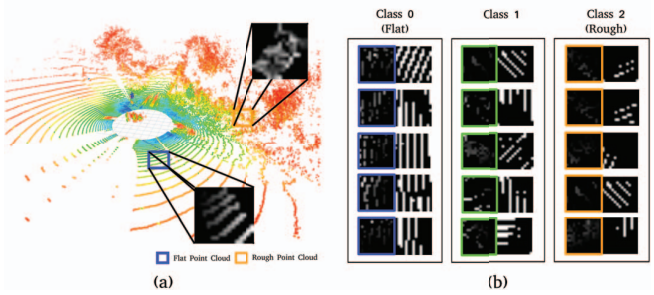

Fig. 4. The process of the Filter-based method for dynamic scenes. (a) Difference in heightmap of each terrain texture. (b) Clustering based on the results of applying the Gabor filter.

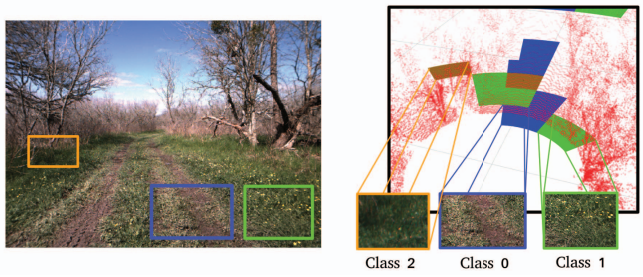

Fig. 5. Comparison of classification results based on terrain texture.

unique terrain forms. Based on this insight, we developed a classification model specifically designed to distinguish these four heightmap classes, providing a complete solution for terrain texture differentiation.

3) Acquire IMU Data with Gazebo Simulator: To understand and analyze the relationship between terrain roughness and IMU data, we turned to empirical studies and simulations. Historically, the relationship between road roughness and IMU data has been scrutinized in various outdoor environments [27]. We utilized the Gazebo simulator with the Warthog robot for our empirical data collection. Heightmap patches were transformed into terrains within the Gazebo world, setting the stage for our experiments. By repeatedly maneuvering the robot across this simulated terrain, we were able to collect IMU data, reflecting varied terrain conditions.

4) IMU Metric: The continuous time-series nature of IMU data contrasts with the discrete snapshots represented by each heightmap patch. To align these domains, we must transform the continuous IMU signal into a quantifiable metric appropriate for patch-wise evaluations. Angular velocity y, denoted as ω_y, became the key metric in our evaluations. This choice was influenced by the fact that the angular velocity y value indicates the oscillation perpendicular to the ground in the direction of the robot's progression. In other words, ω_y serves as a quantifier for the vertical jitters or tremors experienced by the robot as it traverses varied terrains. Drawing inspiration from the International Roughness Index (IRI) [28], we define the IMU cost C_r as:

$$C_r = \frac{\text{Accumulated } \omega_y}{\text{Distance Traveled}}. \tag{1}$$

This formulation considers the robot's movement efficiency. The more the robot wobbles, the shorter the distance it covers in a set timeframe. Dividing the accumulated ω_y by the distance traveled provides a normalized metric. With more jitters, the accumulated ω_y value rises, and the distance traveled diminishes. This makes the C_r value an effective numerical representation of the terrain's roughness. Initial tests affirmed that higher C_r values correlated with rougher terrains.

5) Patch Labelling: In the case of a static scene, the factors affecting terrain roughness are not only the texture of the road but also the difference between the maximum and minimum z values of the patch. Since the same heightmap can represent different terrains if this difference varies, we incorporated the z value difference into the analysis.

By analyzing the distribution of C_r and z value differences in the form of histograms, we demarcated specific intervals. Each patch was subsequently labeled based on these intervals, facilitating the classification process. Our methodology offers a structured approach to evaluating terrain bumpiness and introduces a novel metric, C_r, which enhances the interpretation of IMU data in relation to terrain textures.

C. Dynamic Scene Analysis

In the case of dynamic scenes, this section presents an overview of our methodology that employs a polar grid to generate heightmaps. We then introduce two distinct approaches for evaluating roughness from these heightmaps: the Filter-based method and the Learning-based method.

1) Heightmap Generation: In generating heightmaps in polar grid representation, our methodology addresses a pivotal challenge with 3D LiDAR data : the sparsity issue. As distance from the sensor increases, the point cloud becomes sparse, leading to difficulties in accurate processing. To mitigate this problem, we adopted a polar grid approach and divided the point cloud into multiple bins with varying sizes based on distance from the sensor, as illustrated in Fig 1. Inspired by previous works [25], [29], this method ensures that the farther the sector is from the sensor, the larger it becomes, effectively compensating for the reduction in point density.

2) Traversable Road Selection via Normal Vector Estimation: To reduce the computational cost of assessing roughness for every sector, it was necessary to preliminarily identify

the roads that are potentially traversable by the mobile robot. Therefore, we calculated the normal vector using Principal Component Analysis (PCA) and analyzed the angle of this normal vector to eliminate areas that are completely non-traversable beforehand. However, even if the normal vector is perpendicular to the ground, there are areas that are still non-traversable. These areas are subsequently prevented from being traversed by being classified as "rough roads" in the texture analysis that follows.

To analyze roughness using heightmap patches, we explored two different methods for dealing with dynamic scenes: a Filter-based method and a Learning-based method. The overview of these methods is shown in Figure 6 .

3) Filter-based Method: The Filter-based method leverages the characteristic that, on a flat surface, the 3D LiDAR sensor typically exhibits regular striations. Various angles and frequencies of Gabor filters were applied to the generated heightmap to extract these striation components. Once extracted, the number of striations were counted and subsequently clustered using the k-means++ algorithm, as referenced in [30]. This approach is visualized in Figure 4. The Gabor filter, utilized to detect stripe patterns in grayscale images [31], is expressed as:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \times \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \tag{2}$$

, where

$$x' = x\cos(\theta) + y\sin(\theta), \tag{3}$$

$$y' = -x\sin(\theta) + y\cos(\theta). \tag{4}$$

This formula has been pivotal for extracting the necessary features from the heightmaps.

4) Learning-based Method: The methodology for the Learning-based method involves utilizing the annotation labels of point clouds to create a training dataset. These annotations are semantic, including categories such as mud, grass, and puddles. In each region, the most frequently occurring annotation label is selected as the representative value, forming the heightmap patch train dataset. This dataset is then used to create an image classification model. This model is trained to recognize and classify different ground textures, which is essential for mobile robots navigating in outdoor environments. The accuracy and effectiveness of this method are later validated through experiments and comparisons with other methods.

IV. Experimental Results

The following experiments validate the methods proposed in the methodology using real datasets. Due to their distinct characteristics, static scene and dynamic scene utilized different datasets. Our experimental setup comprised of an Intel Core i7 processor and NVIDIA RTX 3070 GPU.

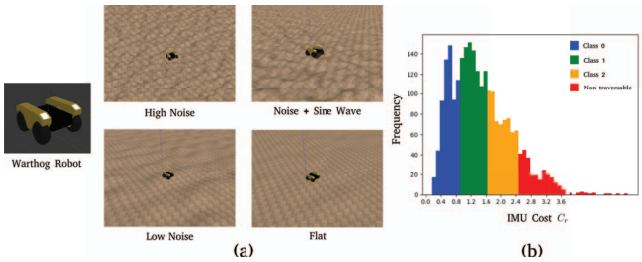

Fig. 6. Terrain analysis with heightmaps and C_r distribution in Warthog simulations.(a) Implementing various types of heightmaps as terrains in the Gazebo simulator and performing repeated experiments with the Warthog robot. (b) Histogram of C_r values obtained through this process.

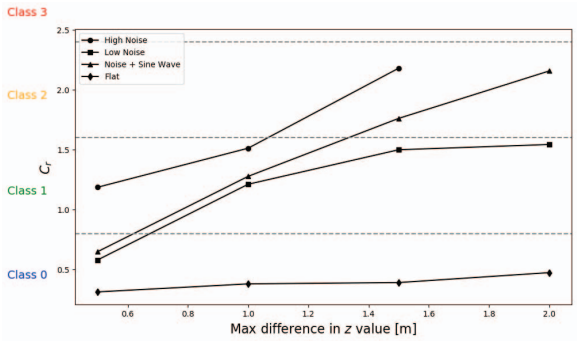

Fig. 7. Average C_r values by terrain type and corresponding class allocation.

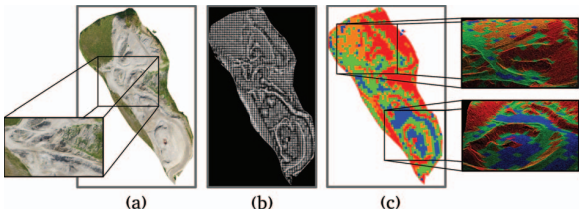

Fig. 8. Process for static scene. (a) Original point cloud, (b) Heightmap patches for each cropped area, (c) Final segmentation result. It can be seen that rough terrain and flat terrain are distinguished at the patch level.

A. Static Scene

1) Dataset and Experimental Setup: The dataset comprises a 3D point cloud obtained using a drone. The Gravel Quarry dataset captures an outdoor area of 0.156 km^2 in Vermont, USA. This locale features a diverse range of rocky terrains encompassing variations like undulations and slopes. Another dataset, Hotel Modelling, originates from an outdoor region spanning 1.19 km^2 in Panama City, Mexico. It incorporates various features such as grasslands, pathways, concrete grounds, and trees. All of these point clouds are publicly available [32]. Each point cloud was cropped into patches of size $3m \times 3m$. These sections were subsequently transformed into heightmap patches, each having dimensions of $15px \times 15px$.

2) Class Segmentation Based on C_r and z values: To determine the segments for final class allocation, considering both C_r values and the difference in z values, we conducted

TABLE I
TERRAIN CHARACTERISTICS FOR EACH CLASS.
(STATIC SCENE)

		Characteristics		
		Variance in z [m²]	Residual in z [m]	Percentage (%)
I	Class 0	0.0039	0.0369	18.24
	Class 1	0.0329	0.0788	12.78
	Class 2	0.0952	0.1718	0.77
	Class 3	36.136	3.6157	68.21
II	Class 0	0.0096	0.036	12.37
	Class 1	0.0693	0.0961	33.26
	Class 2	0.1517	0.0932	17.22
	Class 3	0.6955	0.1759	37.15

following experiments : In a simulation environment constructed with heightmaps in Gazebo simulator, we operated the Warthog robot for 10 seconds, 100 times for each heightmap, and listed a total of 2,478 C_r values acquired from the IMU into a histogram. The histogram was then used to determine the segments for classifying the data, as detailed in Figure 6. As the maximum difference in z value increases, the C_r value tends to rise across different noise types, The *High Noise* consistently maintains the highest C_r values, reflecting its amplified roughnness relative to other terrains, while *Flat* terrain demonstrates the lowest roughness.

3) Neural Network Accuracy and Terrain Characteristics for Each Classification Results: In our endeavor to train the heightmap patch classifier, we utilized a dataset consisting of 26,136 images. Each class within this dataset uniformly contained 6,534 images. For validation purposes, an additional set of 26,136 images was prepared. These were variations of the original heightmaps, modified through brightness adjustments, rotations, and other alterations. Our classifier achieved an accuracy rate of 92.42% on this validation dataset. To ensure that terrains were accurately classified according to their texture, we embarked on a subsequent experiment. The metrics employed for assessment were the variance in z and the residual in z, chosen for their ability to represent the roughness of each patch. To calculate the variance in z, we examined the spread of the z values around their mean for each patch, which gives an insight into the elevation differences within the patch. For the computation of the residual in z, a fitting plane for the respective patch was first identified using the Random Sample Consensus(RANSAC) approach. Following this, the distance between each point in the patch and this plane was calculated. The residual in z was then derived by computing the average of these distances. The variance in z values and the residual in z values for the classified patches are shown in Table I. Table I showcases the results for two datasets: I for the Gravel Quarry dataset and II for the Hotel Modelling dataset. In both datasets, the variance and residual of z values for class 0 are the smallest, denoting the flattest surfaces. These values escalate from class 0 to class 3, with class 3 signifying the roughest surfaces.

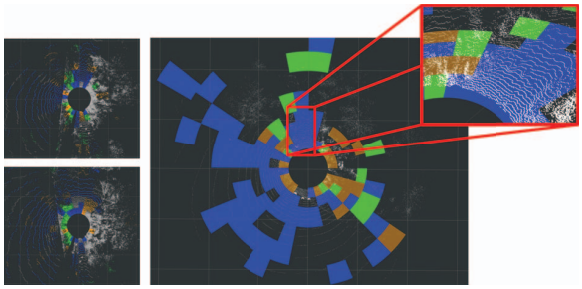

Fig. 9. Top-down view of dynamic scene results.

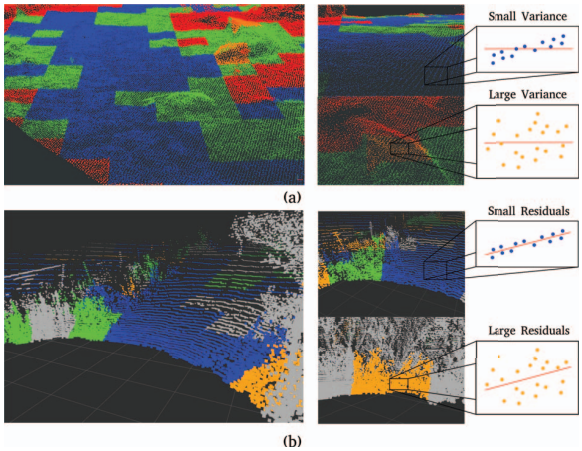

Fig. 10. Terrains segmented by its texture and visualization of evaluation metrics. (a) Detailed images of static scene results and visualization of variances in z. (b) Detailed images of dynamic scene results and visualization of residuals in z.

B. Dynamic Scene

1) Dataset: To evaluate our proposed method's performance, we conducted experiments using the RELLIS-3D dataset [33]. The RELLIS-3D dataset provides high resolution 3D LiDAR data captured in an off-road environment. Given the dataset's unique focus on varied terrains and its high-quality 3D LiDAR scans, it is suited for our experiments aiming to understand and categorize complex outdoor environments.

2) Baseline of Learning-based Method: The Learning-based method utilizes the annotated point cloud dataset from the RELLIS-3D dataset. The RELLIS-3D annotated point cloud dataset provides point-wise annotations for dirt, grass, trees, mud, etc. Based on the criteria presented in Table III, the most dominant label in a given sector was used to classify the heightmap and create the training dataset. The numbers in parentheses next to the items indicate the label numbers referring to the RELLIS-3D dataset. The model used was MobileNetV2 [34], and a total of 2,847 point clouds were used. The training dataset consists of 28,505 instances per class, and the validation dataset consists of 4,077 instances per class. The accuracy for the validation dataset was 83.24%.

TABLE II
COMPARISON OF EXECUTION TIME AND MEMORY USAGE.

	Static Scene				Dynamic Scene				
	Heightmap based method (Ours)	Point cloud based method			Heightmap based method (Ours)		Point cloud based method		
		Variance in z	Residual in z	Total	Filter - based method	Learning - based method	Variance in z	Residual in z	Total
Time (ms / patch)	**12.257**	15.469	25.558	41.027	**3.228**	7.626	6.323	5.941	12.265
Memory (kB / patch)	**0.084**	0.169	1.230	1.399	**3.830**	5.859	9.814	8.301	18.115

TABLE III
CLASS ALLOCATION FOR RELLIS-3D ANNOTATION LABEL.

Class 0 (Flat)	Class 1	Class 2 (Rough)
grass (3)	water(6)	tree (4)
asphalt (17)	puddle (31)	person (17)
concrete (23)	mud (33)	vehicle (8)
	bush (19)	fence (18)
	rubble (34)	barrier (27)

TABLE IV
TERRAIN CHARACTERISTICS FOR EACH CLASS.
(DYNAMIC SCENE : FILTER-BASED METHOD)

Characteristics	Class 0	Class 1	Class 2
Number of stripes	**7.05**	5.03	2.65
Variance in z [m²]	0.00156	0.00403	**0.01070**
Residual in z [m]	0.0173	0.0245	**0.0417**

TABLE V
TERRAIN CHARACTERISTICS FOR EACH CLASS.
(DYNAMIC SCENE : LEARNING-BASED METHOD)

Characteristics	Class 0	Class 1	Class 2
Variance in z [m²]	0.0019	0.0049	**0.0124**
Residual in z [m]	0.0194	0.0318	**0.0507**

3) Terrain Characteristics Based on Classification Results: To evaluate the performance of the Filter-based method and the Learning-based method, we conducted experiments using a total of 2,847 ply files from sequence 00000 provided by RELLIS-3D. These ply files are instant captures of point clouds from the ROSbag file of RELLIS-3D dataset sequence 00000. The variance and residual of z values for patches belonging to each class can be found in Table IV and Table V, respectively.

Analysis of the tables reveals: Class 0 has the most uniform terrains with the smallest z variance and residuals. Meanwhile, Class 2 features the roughest surfaces with the highest z values. The Learning-based method further differentiates terrains with greater clarity, showing a more pronounced difference in variance and residual values between classes compared to the Filter-based method. In summary, while both methods categorize terrains in an increasing order of roughness from Class 0 to Class 2, the Learning-based method proves to be more adept in distinguishing the intricate characteristics of each terrain class.

C. Comparison of Computational Cost Benefits of Using Heightmap

To verify the efficiency of using heightmaps from a computational cost perspective compared to analyzing the point cloud itself, we measured the time and memory usage in both static and dynamic scenes. As before, we used the AgEagle dataset for the static scene and the RELLIS-3D 00000 sequence dataset for the dynamic scene. For the comparison group, the point cloud-based method, we measured the time and memory required to compute the variance in z and residual in z, which are indicators that can represent roughness. The 'Total' refers to the sum of these two values. In both experiments, it was observed that the heightmap based method required less time and memory than the total for the point cloud based method. The details can be found in Table II. The table presented compares the computational costs of heightmap-based methods and point cloud-based methods in terms of execution time and memory usage. Our proposed methods fall under the category of heightmap-based techniques, with all methods further categorized into static and dynamic scenes. Within the static scene category, our heightmap-based method outperforms the point cloud-based method in terms of time efficiency, taking only 12.257 ms/patch compared to the combined time of 41.027 ms/patch of the latter. In terms of memory consumption, our method is more efficient in both scene categories, using only 0.084 kB/patch and 3.830 kB/patch for static and dynamic scenes respectively, which are significantly lower than the totals of the point cloud-based methods.

V. CONCLUSION

In this work, we introduced a novel approach to process 3D LiDAR point cloud data into heightmaps for the identification of less bumpy roads for outdoor robotic navigation. We employed distinct cropping techniques for point clouds in static and dynamic settings, subsequently classifying the resultant heightmaps by texture. In static scenarios, we introduced a new metric, C_r, using IMU data combined with texture analysis outcomes from our heightmap classifier. For dynamic

environments, we processed point clouds into a polar grid, identifying traversable zones using PCA-driven normal vector estimation. Comparative evaluations between the Learning-based method and the Filter-based approach indicated differences in classification performance for heightmaps, with each method having its respective strengths and limitations. Our method consistently evaluated roughness across both scenarios, highlighting its potential to guide mobile robots towards smoother paths. With the rapid growth of outdoor robots, especially in the delivery sector, our study on identifying *less bumpy roads* is crucial for ensuring stability. Future endeavors might explore applying our technique to RGB-D imagery or integrating graph traversal algorithms with our segmentation results to further refine route selection for outdoor navigation.

ACKNOWLEDGMENT

This research was funded by the Ministry of Trade, Industry and Energy (MOTIE), South Korea, under Industrial Technology Innovation Program under Grant 20015440, and partly supported by the BK21 plus program "AgeTech-Service Convergence Major" through the National Research Foundation (NRF) funded by the Ministry of Education of Korea[5120200313836], and by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2022-00155911, Artificial Intelligence Convergence Innovation Human Resources Development (Kyung Hee University)).

REFERENCES

[1] L. Bruzzone and G. Quaglia, "Review article: locomotion systems for ground mobile robots in unstructured environments," *Mechanical Sciences*, vol. 3, no. 2, pp. 49–62, 2012.

[2] S. Shentu, Z. Gong, X.-J. Liu, Q. Liu, and F. Xie, "Hybrid navigation system based autonomous positioning and path planning for mobile robots," *Chinese Journal of Mechanical Engineering*, vol. 35, p. 109, Aug 2022.

[3] K. Berns, K.-D. Kuhnert, and C. Armbrust, "Off-road robotics—an overview," *KI - Künstliche Intelligenz*, vol. 25, pp. 109–116, May 2011.

[4] N. Bouton, R. Lenain, B. Thuilot, and P. Martinet, "A new device dedicated to autonomous mobile robot dynamic stability: Application to an off-road mobile robot," in *2010 IEEE International Conference on Robotics and Automation*, pp. 3813–3818, 2010.

[5] A. Howard and H. Seraji, "Vision-based terrain characterization and traversability assessment," *Journal of Robotic Systems*, vol. 18, no. 10, pp. 577–587, 2001.

[6] D. Silver, J. A. D. Bagnell, and A. T. Stentz, "Learning from demonstration for autonomous navigation in complex unstructured terrain," *International Journal of Robotics Research*, vol. 29, pp. 1565 – 1592, October 2010.

[7] G. G. Waibel, T. Löw, M. Nass, D. Howard, T. Bandyopadhyay, and P. V. K. Borges, "How rough is the path? terrain traversability estimation for local and global path planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16462–16473, 2022.

[8] L. Li and C. Sandu, "On the impact of cargo weight, vehicle parameters, and terrain characteristics on the prediction of traction for off-road vehicles," *Journal of Terramechanics*, vol. 44, no. 3, pp. 221–238, 2007.

[9] K. Viswanath, K. Singh, P. Jiang, S. P. B., and S. Saripalli, "Offseg: A semantic segmentation framework for off-road driving," 2021.

[10] A. Kurup, S. Kysar, and J. Bos, "Svm based sensor fusion for improved terrain classification (conference presentation)," p. 16, 04 2020.

[11] B. Suger, B. Steder, and W. Burgard, "Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3d-lidar data," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3941–3946, 2015.

[12] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, and K. Schwehr, "Recent progress in local and global traversability for planetary rovers," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, pp. 1194–1200 vol.2, 2000.

[13] N. E. Budiyanta, E. M. Yuniarno, T. Usagawa, and M. H. Purnomo, "Normal vector direction-based 3d lidar point cloud planar surface removal for object cluster minimization in human activity monitoring system," in *2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, 2023.

[14] M. Oh, E. Jung, H. Lim, W. Song, S. Hu, E. M. Lee, J. Park, J. Kim, J. Lee, and H. Myung, "Travel: Traversable ground and above-ground object segmentation using graph representation of 3d lidar scans," 2022.

[15] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud," 2022.

[16] M. A. Bekhti and Y. Kobayashi, "Regressed terrain traversability cost for autonomous navigation based on image textures," *Applied Sciences*, vol. 10, no. 4, 2020.

[17] R. Kumar, P. Kulashekar, B. Dhanasekar, and B. Ramamoorthy, "Application of digital image magnification for surface roughness evaluation using machine vision," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 2, pp. 228–234, 2005.

[18] D. Stavens and S. Thrun, "A self-supervised terrain roughness estimator for off-road autonomous driving," 2012.

[19] M. H. Bharati, J. Liu, and J. F. MacGregor, "Image texture analysis: methods and comparisons," *Chemometrics and Intelligent Laboratory Systems*, vol. 72, no. 1, pp. 57–71, 2004.

[20] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[21] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 1, pp. 25–39, 1983.

[22] F. Bianconi and A. Fernández, "Evaluation of the effects of gabor filter parameters on texture classification," *Pattern Recognition*, vol. 40, no. 12, pp. 3325–3335, 2007.

[23] F. Riaz, A. Hassan, S. Rehman, and U. Qamar, "Texture classification using rotation- and scale-invariant gabor texture features," *IEEE Signal Processing Letters*, vol. 20, no. 6, pp. 607–610, 2013.

[24] V. Andrearczyk and P. F. Whelan, "Using filter banks in convolutional neural networks for texture classification," *Pattern Recognition Letters*, vol. 84, pp. 63–69, 2016.

[25] H. Lim, O. Minho, and H. Myung, "Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3d lidar sensor," *IEEE Robotics and Automation Letters*, 2021.

[26] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, "Learning ground traversability from simulations," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1695–1702, jul 2018.

[27] W. Wen, "Road roughness detection by analysing imu data," 01 2008.

[28] W. D. Paterson, "International roughness index: Relationship to other measures of roughness and riding quality," *Transportation Research Record*, 1986.

[29] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3d laser scans for online operation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 163–169, 2016.

[30] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," vol. 8, pp. 1027–1035, 01 2007.

[31] U. Marmol, "Use of gabor filters for texture classification of airborne images and lidar data," *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, vol. 22, 2011.

[32] A. A. S. Inc., "Ageagle dataset : Gravel quarry and hotel modelling."

[33] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," 2022.

[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019.