2025 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-4139-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICRA55743.2025.11127990

Surface Roughness Estimation for Terrain Perception

Minxiang Ye[1,3], Yifei Zhang[2,3], Jason Gu[4], Senwei Xiang[2], Lingyu Kong[3], and Anhuan Xie[1]

Abstract— **Ground terrain perception has become the primary visual task for the robust navigation of intelligent systems in unstructured outdoor environments. However, complex terrain poses a significant challenge to vision-based perception. This work introduces a novel estimation task using RGB images to facilitate low-cost terrain perception in extracting surface roughness information. The proposed task presents both semantic-aware and edge-aware roughness descriptors at the pixel level instead of a single value for a given image. To promote the research on the proposed novel terrain roughness estimation task, we introduce a multimodal synthetic dataset for terrain perception in outdoor scenes, containing multiple terrain categories, diverse viewpoints, different lighting and weather conditions, as well as semantic and roughness annotations. Additionally, inspired by computer graphics, we introduce TRENet, a roughness estimation architecture to model the intrinsic correlation of depth-normal-roughness. We also perform ablation studies on the effect of each component and diverse types of inputs. Extensive evaluations and comparisons demonstrate that our method can effectively predict pixel-wise terrain surface roughness with high accuracy.**

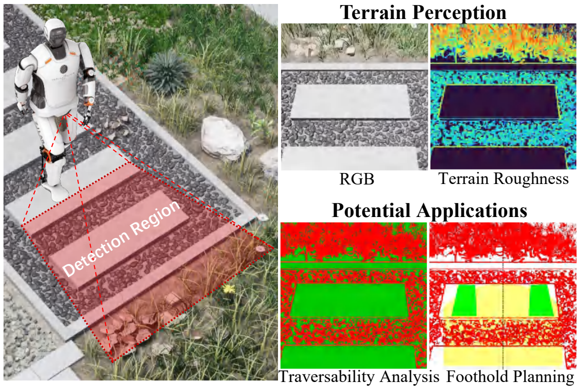


Fig. 1: **Proposed terrain roughness estimation (TRE) task for terrain perception and potential applications in robotics.** Red and green in Traversability Analysis represent the danger and safe zone for robot navigation, respectively. Yellow and green in Foothold Analysis mean valid region and visual-based foothold selection for a biped robot.

I. Introduction

Visual terrain perception is attracting significant attention in robotics. One major reason is that unstructured complex scenes pose great challenges for path planning, traversability analysis, and autonomous control of robotic systems. Generally, intelligent robot systems need to make control decisions based on complete terrain information, such as speed adjustments [1], transformation [2], and reconfigurable path-planning [3], due to their different scene adaptability. Related applications are widely used in logistics [4], search and rescue [5], agriculture [6], etc. Especially for legged robots, improving visual terrain recognition can reduce the difficulty of balance control, thereby achieving robust and safe navigation [7], [8].

With the development of sensor technology, various cameras, such as monocular and stereo cameras, are integrated into robotic visual perception systems [9], [10]. LiDAR is also widely used to obtain spatial information in complex scenes. The height map of scenes can be built for path planning by aggregating point clouds [11], [12]. However, since LiDAR detects objects by emitting lasers, its performance is greatly affected by the environment, such as rain and dust. In recent years, perception systems have increasingly adopted multi-modal sensor fusion for environmental perception [13], [14]. That is, through the feature acquisition of visual cameras in terms of color, texture, semantics, etc., combined with the precise spatial information, the complementation of information in different dimensions can be achieved [15], making the environmental perception more stable.

Advances in deep learning techniques have driven vision-based terrain recognition (e.g. detection, classification, and semantic segmentation), resulting in the emergence of a large number of efficient algorithms [9], [16]. Although these perception tasks have been extensively studied in computer vision, current robot navigation and autonomous control have yet to fully benefit from the advances. Namely, existing terrain perception methods do not sufficiently consider the potential impact of terrain surface on robot navigation and balance control. The traversability analysis and stabilization control of a mobile robot is not only derived from semantic information but also from the geometric layout of object edges in the scene. Extracting roughness information of terrain surface is more practical and meaningful for robots, especially for legged robots. Prior research [17] has investigated the possibility of using a self-supervised terrain roughness estimator for autonomous off-road driving. Advanced robot systems, such as the centaur-like wheeled-legged robot CENTAURO [18], have attempted to segment terrains and estimate their roughness for adaptively controlling the hip motor. Current roughness estimation algorithms are still relatively limited in fine-grained terrain perception, due to

*Yifei Zhang is the corresponding author (yifei.zhang.fr@gmail.com)

*This work is supported by the National Natural Science Foundation of China (Grants No.62201518, 62103383), ”Pioneer” and ”Leading Goose” RD Program of Zhejiang (No.2023C01177).

[1]Haichuang Humanoid Robot Industry Innovation Center, Hangzhou.
[2]Hangzhou International Innovation Institute, Beihang University, 311115, China.
[3]Zhejiang Lab, Hangzhou, 311121, China.
[4]Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS Canada B3H 4R2.

differences in terrain adaptability of robot systems [19].

The above facts motivate us to redetermine the terrain roughness estimation task (sometimes referred to as terrain unevenness) and propose a novel learning-based method and corresponding benchmark dataset. Unlike previous work, we define terrain roughness estimation as a regression problem, which is both semantic-aware and edge-aware. We aim to predict terrain roughness by extracting color, texture, and geometric information from vision-based input. As shown in Fig. 1, the roughness prediction can be potentially used to combine visual terrain perception with robot control, such as in biped robot walking [20] and locomotion using deep reinforcement learning [21].

In summary, the main contributions of this work are: **(1)** A synthetic dataset of unstructured outdoor scenes (RougE), covering multiple terrain categories, diverse viewpoints, various lighting and weather conditions which provides more accurate scene information and multimodal data compared with other real-world datasets for terrain recognition. **(2)** An automatic ground-truth annotation tool, which greatly simplifies the research on terrain roughness estimation by our proposed pixel-wise terrain roughness extraction method. With this tool, we also provide a new dataset, Hypersim-R. **(3)** A learning-based terrain roughness estimation method with great flexibility and interpretability. Ablation studies and extensive analysis of different conventional and CNN-based baseline methods demonstrate the effectiveness of the proposed method.

II. Related Work

Terrain recognition. Terrain recognition has received considerable attention in robot perception, control, and motion planning. Especially for complex unstructured outdoor scenes, visual terrain perception is the premise of realizing traversability assessment and safe navigation for mobile robots. Previous studies of terrain recognition mainly focus on the classification of terrain categories [22]. Combined with prior human knowledge, terrain classification allows for further inference of traversable areas. With the development of computer vision, researchers turned more attention to deep learning-based terrain segmentation [23], [24]. Another line of the solution is to recognize the material properties such as friction[25] or slippage [26]. These properties may require not only appearance but also geometric properties or acoustics to be determined. [27] proposed constructing an arbitrary shape landscape in Gazebo from image input, involving adjusted terrain information of height, scale, rotation angle, sharpness, and texture. Then the terrain relief can be marked as a height map. The most related work to ours is that of [18], which proposed an RGB-based deep neural network method to predict terrain classes and regress the roughness.

Monocular depth estimation. Deep learning techniques have been proven to be effective methods for depth estimation. Although the CNNs have performed powerful feature extraction capabilities, a single RGB image can only provide texture and appearance without direct 3D geometric information, which poses a great challenge to vision-based depth estimation. In recent years, studies on monocular depth estimation have emerged in large numbers, such as [28]. Notably, the work in GeoNet++ [29] focused on depth-normal consistency and its geometric constraints, simultaneously improving depth and surface normal estimation from a single image. Several works [30] tried to use one to help improve the other or allow for multi-task learning.

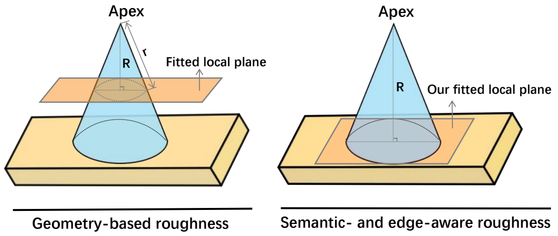


Fig. 2: An example to show the difference between the proposed method and existing method [18] for terrain roughness estimation. R means terrain roughness. r represents the radius of the considered spatial extent.

Surface normal estimation. Surface normals represent important properties of geometric surfaces and are the basis for 3D modeling and surface reconstruction. To name a few, [31] incorporated local, global, and vanishing point information in the network, predicting surface normal from a single image. [32] introduced a skip-connected architecture to fuse hidden representations of different layers for surface normal estimation. Driven by the demands from robotics and autonomous driving, normal estimation was further expanded to the ground plane normal estimation task. [33] proposed GroundNet to estimate the 3D orientation of the ground plane, which is an end-to-end monocular estimation network with geometric consistency loss. Additionally, some work attempts to estimate a set of tactile physical properties of surfaces from visual information [34].

Our work benefits from the existing depth and normal estimation methods. However, these methods do not adequately consider the impact of terrain perception in robotics. This motivates us to further explore the geometric relationship between depth, surface normal, and roughness, then redetermine terrain roughness estimation with semantic-aware and edge-aware annotations.

III. Terrain Roughness Estimation Task and Dataset

A. *Task format*

The format for terrain roughness/unevenness estimation is a regression problem. The task aims to obtain a value $r \in \mathbb{R}$ that represents the distance to the corresponding local plane for each pixel in the image space. Besides, different from geometry-based roughness/unevenness estimation, the proposed task is also semantic-aware and edge-aware (Fig. 2). The inputs can be diversified, *i.e.* in this work, we mainly focus on vision-based image input that provides low-cost terrain perception ability in the outdoor environment.

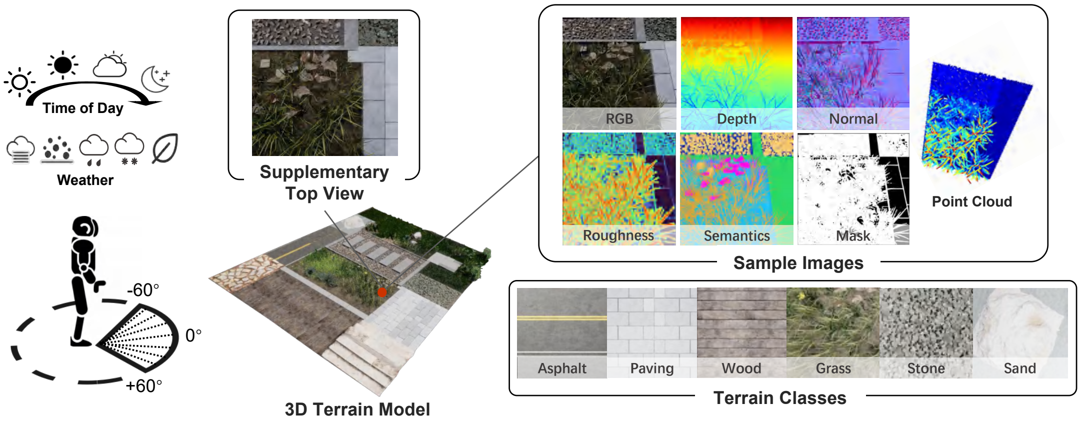


Fig. 3: **Overview of the proposed RougE dataset.** The left area presents the terrain modeling and data collection. The right area shows samples of captured multimodal images, as well as multiple terrain classes in our dataset. This dataset focuses on outdoor terrain scenes and supports a variety of low-level and high-level computer vision tasks, such as depth, surface normal, roughness estimation, semantic segmentation, and 3D reconstruction.

Dataset	**Scale** Train	 Test	 Full	**Shape**	**Multi-view**	**Scene**
RougE	2,688	672	10,752	512 × 512	✓	Outdoor
Hypersim-R	5,157	2,580	77,400		-	Indoor

TABLE I: Dataset statistics.

B. Datasets for TRE task

1) RougE Dataset: We construct a first-of-its-kind synthetic dataset for terrain perception in unstructured outdoor scenes. The dataset contains accurate and reliable data for multiple modalities and can support a variety of low-level and high-level computer vision tasks.

Camera setup. In the Unreal Engine 4 virtual environment, we create 9 virtual ideal cameras (1280 × 720) at the same world coordinates with an angular separation of 15° from -60° to 60° (Fig. 3). To simulate the terrain perception system for robot navigation, we set the virtual cameras to be 1 meter high on the z-axis and look down at 60°.

Dataset collection. We provide 6 different terrain categories with high-quality mesh. Various light sources, such as point light, line light, and natural light, are utilized to enrich the feature representation of RGB images. Finally, we construct a high-quality multimodal synthetic dataset (**RougE**) with 4 modalities, *i.e.* RGB images, depth maps, surface normal, point cloud, semantic labels, terrain roughness, semantic and edge-aware mask in 4 lighting conditions, 5 weather conditions, 9 viewing angles and a real-world test subset.

2) Hypersim-R Dataset: To further validate the proposed TRE task and methods, we employ the proposed terrain roughness annotation tool on Hypersim dataset [35] to generate a new dataset, **Hypersim-R**. The original photorealistic synthetic dataset Hypersim was created for holistic indoor scene understanding. Focusing on robotics settings similar to **RougE**, we select those multi-modal image pairs that are looking down $[0°, 45°]$ and are captured $[0.3, 1.2]$ meters above the ground floor. The resulting data statistics and comparisons can be found in Tab. I.

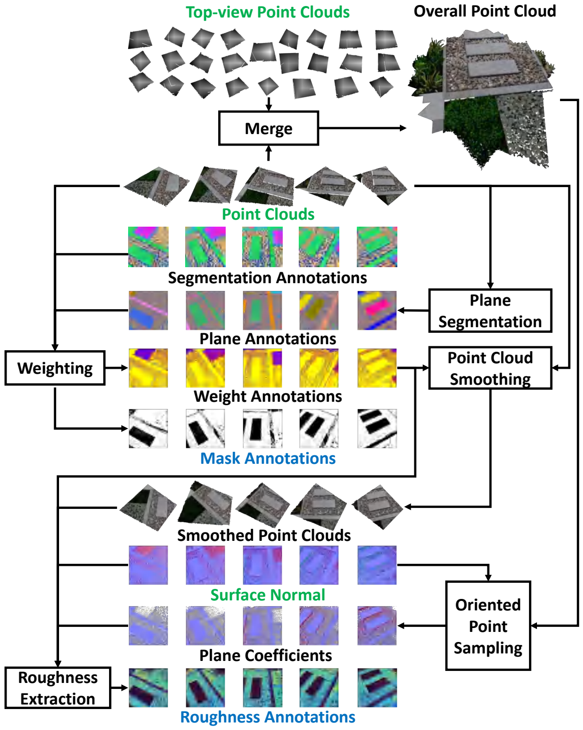


Fig. 4: **Proposed terrain roughness generation.**

C. Terrain roughness ground-truth

Current research in extracting surface roughness information remains limited for robotics terrain perception. In [18], 3D depth data are firstly generated by Structure from Motion (SfM) from a collection of terrain images in the GeoMat dataset [36]. The resulting point clouds associated with each terrain image are then used to generate the roughness value r_i for each point (x_i, y_i, z_i):

$$r_i = \frac{|-d_i - a_i x_i - b_i y_i - c_i z_i|}{\sqrt{a_i^2 + b_i^2 + c_i^2}}, \tag{1}$$

where the local plane $(a_i x_{\mathcal{N}_i} + b_i y_{\mathcal{N}_i} + c_i z_{\mathcal{N}_i} + d_i = 0)$ for i-th point is fitted using the least squares method [37] over its neighboring point sets $\mathcal{N}_i$. Existing work focus particularly on the relatively local region around each pixel. Accordingly, neighbor selection methods are critical for defining such local regions in order to better represent the physical informed characteristics, especially the distances between the adjacent planes that are associated with different objects. To facilitate terrain perception in understanding the geometric layout of smooth-riding surfaces, this work proposes a novel pixel-wise roughness extraction method that takes both semantic and edge information into account.

Given a coordinate index (x, y) in the virtual environment, the process of terrain roughness generation is performed over all view angles in parallel (Fig.4). To explicitly represent the distances between the adjacent planes that are associated with different objects, we generate the weight and mask annotations that take the point cloud, segmentation annotations, and plane annotations into account. In detail, the mask image annotates the binary conditions of whether the pixel is at the object/plane boundary or not, where plane annotations are extracted using organized point cloud segmentation with connected components [38]. For weight image annotations, each pixel value is set to the maximum Euclidean distance over its 8-neighbors when the pixel is located at either the object or plane boundaries. Otherwise, the value is set to zero when either the boundary condition is not satisfied or the maximum Euclidean distance is smaller than the geometric resolution of $\epsilon = 1.5$cm. The resulting weight annotations are used to adaptively smooth the point for providing smooth transients near the object/plane boundaries, such that the final roughness value can better reflect the geometric edges.

To better capture the intricate geometric details, four-neighboring top-view point clouds over all view angles and the point clouds at (x, y) are merged into an overall point cloud. Given the surface normal generated by the render engine, the overall point cloud is then used to estimate the local plane coefficients d_i for i-th point within the point clouds at the corresponding image coordinate index (x, y) via oriented point sampling [39]. Eventually, the terrain roughness pixel value can be calculated by Eq.1 using the smoothed point clouds, surface normal, and plane coefficients. To recover the roughness value when the distance between the object/plane boundary is smaller than ϵ, its value is multiplied by 2 if its weight annotation is larger than 0.5ϵ.

IV. Methodology

A. Overview

Fig.5a depicts the overall structure of our TRENet. The system pipeline consists of the following components: **(1)** learning-based depth estimator: we adopt a light-weight ResUnet-like regression network to perform the depth estimation, **(2)** depth-to-pointcloud transformation module: the depth maps are firstly inversely reprojected to the downside view angle $\beta = 60^\circ$ and then transformed into a 3D point cloud using the pinhole camera model, **(3)** DSAC-based neighbor selector: we employ a conventional residual convolutional network to generate the probabilities ρ_i of 7×7 neighbors and provide robust neighboring hypotheses, **(4)** surface normal estimator: given a point cloud and the corresponding neighboring hypotheses, we adopt the existing parallel differentiable least-square method [40] on the hypotheses for normal estimation, **(5)** surface normal refinement module: following [29], we refine the initial estimated normal vectors to match the ground-truth surface normal via the same architecture as (1), **(6)** edge-aware surface reconstruction module: given the transformed point cloud, a conventional residual convolutional network is learnt to extract the centroids of each point in the refined point cloud and reconstruct the plane coefficients, **(7)** roughness estimator: we calculate the terrain roughness value using Eq.1 and the reconstructed plane coefficients.

For the residual block in the GeoNet-R architecture shown in Fig.5b, we stack two convolutional blocks and apply a residual skip connection between the overall input and output. Each convolutional block consists of a 2D convolution layer with a 3×3 kernel size, a batch normalization layer, and a leaky rectified linear unit with a negative slope of 0.2. One can also formulate the TRE task as an end-to-end regression problem that relies on a learning-based roughness estimator. Unlike the end-to-end CNN-based roughness estimation pipeline, TRENet retains interpretability and provides the possibility of replacing the depth estimator using sensing technology, such as stereo imaging, LIDAR, etc..

B. RANSAC-based plane fit

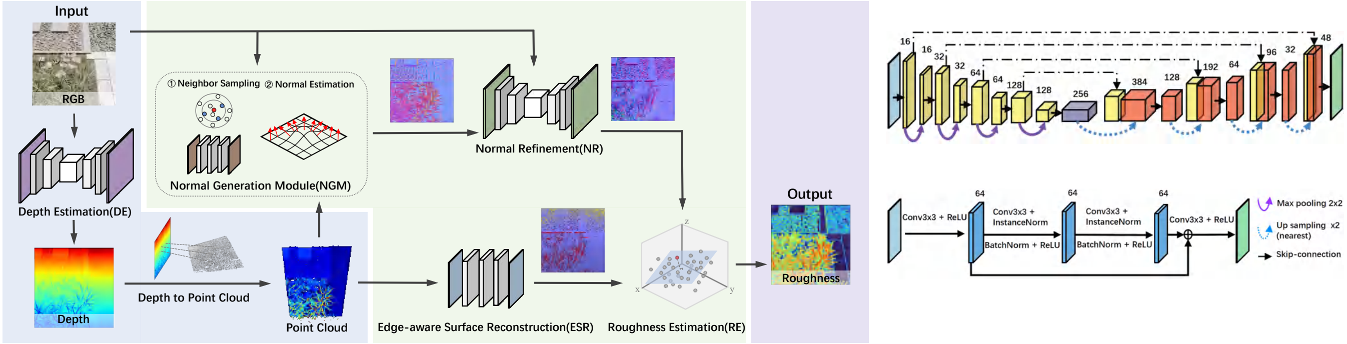
Surface normal estimation for i-th point in a point cloud can be formulated as estimating a normal vector N_i whose direction matches the corresponding actual tangent plane normal. The least-square (LS)-based plane fitting is one of the standard methods to estimate the surface normal at the i-th point. However, the LS method is not robust to outliers and does not generate high-quality surface normal [41]. RANSAC is an alternative method to reduce the effect of the outliers. Motivated by [40], we adopt the differentiable RANSAC (DSAC) to perform the plane fitting using the soft-argmax based hypothesis selection Eq.2.

$$\begin{aligned} h_{DSAC}^{w,v} &= h_J^w, \text{ with } J \sim P(J \mid v, w), \\ P(J \mid v, w) &= \frac{\exp(s(h_J^w, Y^w; v))}{\sum_{J'} \exp(s(h_J^w, Y^w; v))}, \end{aligned} \tag{2}$$

where parameters w influence the quality of competing hypotheses h but do not influence the initial uniform sampling of minimal sets Y_J. A network v is employed to generate the probabilities ρ_i of 7×7 neighbors for the probabilistic selection. Following [40], we generate three indices for each point around its neighbors based on a multinomial distribution ρ_i, avoiding the underdefined plane fitting problem. DSAC generates a hypothesis with more inliers on the actual tangent plane, resulting in better normal vectors than LS.

C. Edge-aware surface reconstruction

Given a point cloud $\mathcal{V}$ whose i-th point is denoted by (x_i, y_i, z_i), the edge-aware surface reconstruction is performed to refine the reconstructed local surface plane



(a) **TRENet scheme.** TRENet can take a monocular RGB image as the input to estimate its depth maps at first. The point cloud can be computed via the pinhole camera model. Normal estimation is performed using neighbor sampling. Following a normal refinement module, it corrects the surface normal. The proposed roughness estimation module is eventually adopted to measure the terrain roughness based on the reconstructed local plane coefficients. Note that the input sources are flexible in this scheme.

(b) **Sub-network in TRENet.** A lightweight ResUnet-like network (GeoNet-R) is used for depth estimation and normal refinement module in TRENet. We adopt the second conventional residual convolutional network in the neighbor sampling and point cloud refinement modules.

Fig. 5: **Overview of the proposed TRENet.**

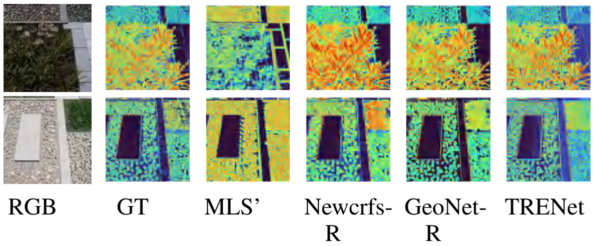


Fig. 6: Visual comparison between all baselines.

$a_i x_{\mathcal{N}_i} + b_i y_{\mathcal{N}_i} + c_i z_{\mathcal{N}_i} + d_i = 0$ such that the estimated terrain roughness matches the ground-truth $\hat{r}_i$, formulating:

$$g = \arg\min_{g} |\hat{r}_i - \mathcal{N}_i * \mathcal{Q}(x_i, y_i, z_i; g)| \tag{3}$$

where $\mathcal{Q}(x_i, y_i, z_i; g)$ is a function that extracts the centroid of i-th point in the reconstructed local plane based on the parameter set g. In this work, we employ a CNN-based network with weights g that estimates the pixel-wise centroids for the reconstruction task. The plane coefficient d_i for the reconstructed plane can be used to calculate the final roughness value via Eq.1.

V. Experiments

A. Experimental setup

Given RGB images, we initialize the front-end depth estimation module with a GeoNet-R network [29] pre-trained on the topview subset. Adam optimizer [42] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$ is used to optimize all networks. The initial learning rate is set to $1e{-}3$ and adjusted following a decay of 0.5 every 50k iterations over 200k iterations in total. The batch size is set to 12, 4 and 1 for GeoNet-R, Newcrfs-R and TRENet, respectively. 12 hypotheses of neighbors are sampled for training the DSAC-based normal estimation module. For depth estimation, following the prior works [43], [44], we use the L1 and SSIM loss with the weights of 1.0 and 0.5, respectively. For surface normal estimation, we adopt the consistency loss to minimize the angle between the predicted surface normal vector and its ground truth. L1 loss is simply used to train the roughness estimation module in the proposed TRENet.

Method	Input	RMSE	B-RMSE	F-RMSE
			Lower is better	
MLS' [45]	PointCloud	2.882	3.674	0.816
Newcrfs-R [46]	RGB	1.706	2.168	0.315
GeoNet-R [29]	RGB	1.522	1.931	0.299
TRENet	RGB	**1.514**	**1.926**	**0.271**

TABLE II: Performance evaluation of terrain roughness estimation on **RougE** dataset.

Baselines. To better capture the geometric neighboring information, the point cloud captured by the virtual camera is densified via the moving least squares surface (**MLS'**) reconstruction using [45]. The searching radius is set to 1cm for MLS', with a second-order polynomial fitting operator. The local plane of each input point is upsampled in a circular fusion with a radius of 0.01 and a step size of 0.01. The surface roughness computation is then conducted on the resulting reconstructed point cloud using Eq. 1. We compare two end-to-end SOTA depth estimation methods (*i.e.* **GeoNet** [29] and **Newcrfs** [46]), in which we make slight modifications to adopt the terrain roughness estimation task (annotated with -R).

Metrics. Similar to the depth estimation tasks [29], [47], we evaluate the performance of terrain roughness estimation using: (1) root mean square error (**RMSE**), (2) boundary root mean square error on the boundary pixels (**B-RMSE**), (3) flat root mean square error on the boundary pixels (**F-RMSE**). The boundary pixels are masked as "white" in mask annotations M during the ground-truth generation (Sec. III-C), while the flat pixels are the opposite condition.

B. Quantitative results

Results on RougE dataset. We first conduct experiments on a subset of **RougE** dataset with different lighting conditions. As shown in Tab. II, the proposed TRENet pipeline achieves

Method	Input	RMSE	B-RMSE	F-RMSE
		Lower is better		
MLS' [45]	PointCloud	58.590	103.525	32.865
Newcrfs-R [46]	RGB	60.620	101.682	33.050
GeoNet-R [29]	RGB	55.194	92.261	29.390
TRENet	RGB	51.781	84.940	26.446

TABLE III: Performance evaluation of terrain roughness estimation on **Hypersim-R** dataset.

Module	Input	RMSE	B-RMSE	F-RMSE
		Lower is better		
DE+NGM+NR	RGB	2.089	2.612	0.666
DE+NGM+ESR		1.523	1.940	0.276
DE+LS+NR+ESR		1.721	2.192	0.297
DE+LS+ESR		1.674	2.129	0.297
NGM+NR	PointCloud +RGB	2.221	2.795	0.555
NGM+ESR		1.350	1.723	0.247
LS+NR+ESR	PointCloud	1.408	1.786	0.201
LS+ESR		1.364	1.736	0.218
NGM+NR	DepthSensor +RGB	2.595	3.276	0.554
NGM+ESR		1.920	2.437	0.460
LS+NR+ESR	DepthSensor	2.015	2.559	0.454
LS+ESR		2.071	2.629	0.450
DE+NGM+NR+ESR	RGB	1.514	1.926	0.271
NGM+NR+ESR	PointCloud +RGB	1.347	1.710	0.258
NGM+NR+ESR	DepthSensor +RGB	1.904	2.417	0.453

TABLE IV: Ablation studies on **RougE** test set for Terrain Roughness Estimation.

overall better performance than the end-to-end learning-based Newcrfs-R and GeoNet-R approach. The TRENet outperforms other methods in RMSE and F-RMSE.To make a more intuitive comparison, we visualize two examples in Fig. 6. It is obvious that our method performs better in predicting the details of terrain surface in the scene.

Results on Hypersim-R dataset. To further verify the effectiveness of the proposed terrain roughness estimation method, we conducted experiments on **Hypersim-R** dataset. Note that we adopt the work in [48] for depth estimation. We train the pretrained depth estimation model with **Hypersim-R** train set for fine-tuning, and other modules are configured the same as in the previous experiments. Tab. III shows that TRENet outperforms the baselines in indoor scenes.

C. Ablation study

As shown in Tab. IV, it can be seen that removing any one of the components would lead to a performance drop for RGB input, especially without edge-aware surface reconstruction (ESR). Besides, we conduct ablation studies with different input. It is obvious that the TRENet with point cloud and RGB inputs achieves the optimal performance, reaching 1.347 on RMSE. We also present the ablation studies with different modules and input for normal estimation, which is an intermediate task in this work. Tab. V shows that both NGM+NR and LS+NR perform outstandingly in Error and Accuracy, but in terms of Time, NGM+NR used in our network is significantly better than LS+NR. Note that the input of DepthSensor mentioned in Tab. IV and Tab. V are taken from the work of [49], [50], in which the depth data is estimated by stereo images.

Module	Input	*Error*			*Accuracy*			*Time*
		Mean	**Median**	**RMSE**	**11.25°**	**22.5°**	**30°**	(*ms*)
LS	RGB	45.6	17.4	66.7	10.4	56.9	62.1	103.0
	PointCloud	43.2	9.6	74.9	52.7	64.6	69.0	
	DepthSensor	41.8	19.8	58.2	14.5	54.6	61.3	
NGM	RGB	40.9	21.6	55.2	12.0	50.9	57.0	8.7
	PointCloud	32.8	8.5	55.9	54.4	64.8	69.0	
	DepthSensor	38.2	20.1	51.7	14.8	54.1	61.6	
NGM+NR	RGB	12.8	4.9	24.0	69.9	83.8	88.0	13.2
	PointCloud +RGB	10.8	4.2	21.2	74.9	87.7	91.0	
	DepthSensor +RGB	12.9	4.9	24.3	69.9	83.7	87.9	
LS+NR	RGB	12.8	4.9	24.0	69.9	83.7	88.0	112.7
	PointCloud +RGB	10.9	4.3	21.2	74.6	87.7	91.0	
	DepthSensor +RGB	12.9	4.9	24.4	70.0	83.7	87.8	

TABLE V: Ablation studies on **RougE** test set for Normal Estimation Task.

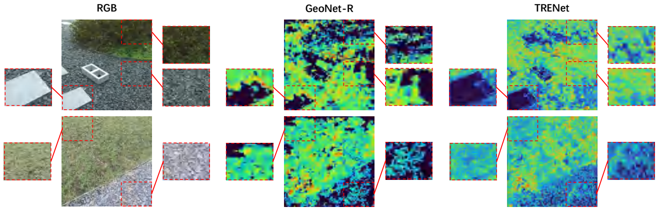


Fig. 7: Qualitative analysis of real-world experiments.

D. Real world experiments

To further show the effectiveness of our method, we evaluate different baseline models on the real-world test set. All the real-world RGB images are captured by a ZED2i camera in a park scene. Fig. 7 demonstrates the terrain roughness estimation of GeoNet-R and our proposed TRENet. Notably, we marked the failure cases with red boxes, and local features are enlarged to compare the performance of different methods on the perception of terrain roughness in real-world scenes. We can observe that GeoNet-R may fail to predict edge regions and details, and performs worse than our method on all the categories.

VI. Conclusion

This work has proposed a novel terrain roughness estimation task that is both semantic-aware and edge-aware. A high-quality multi-modal dataset with semantic labels, terrain roughness, semantic and edge-aware masks was provided for benchmarking terrain roughness estimation methods. Furthermore, we have presented a flexible and interpretable roughness prediction method based on the heuristic combination of deep neural networks and computer graphics. We have conducted several experiments and ablation studies on the proposed dataset, demonstrating the effectiveness of the proposed method. In our future work, we will further expand the scene and dataset, and explore the combination of terrain roughness with robot navigation and control.

References

[1] M. Castelnovi, R. Arkin, and T. R. Collins, "Reactive speed control system based on terrain roughness detection," in *ICRA*, 2005.

[2] F. Masataka, R. E. Mohan, N. Tan, A. Nakamura, and T. Pathmakumar, "Terrain perception in a shape shifting rolling-crawling robot," *Robotics*, vol. 5, no. 4, p. 19, 2016.

[3] A. Chilian and H. Hirschmüller, "Stereo camera based navigation of mobile robots on rough terrain," in *IROS*, 2009.

[4] AgilityRobotics, "Digit & cassie," 2019, uRL: https://www.agilityrobotics.com/robots.

[5] R. Ventura and P. U. Lima, "Search and rescue robots: The civil protection teams of the future," in *2012 Third International Conference on Emerging Security Technologies*. IEEE, 2012, pp. 12–19.

[6] T. Duckett, S. Pearson, S. Blackmore, B. Grieve, W.-H. Chen, G. Cielniak, J. Cleaversmith, J. Dai, S. Davis, C. Fox *et al.*, "Agricultural robotics: the future of robotic agriculture," *arXiv preprint arXiv:1806.06762*, 2018.

[7] J. Frey, D. Hoeller, S. Khattak, and M. Hutter, "Locomotion policy guided traversability learning using volumetric representations of complex environments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 5722–5729.

[8] S. Sato, Y. Kojio, Y. Kakiuchi, K. Kojima, K. Okada, and M. Inaba, "Robust humanoid walking system considering recognized terrain and robots' balance," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8298–8305.

[9] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng *et al.*, "Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation," *arXiv preprint arXiv:2303.15771*, 2023.

[10] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Conference on Robot Learning*. PMLR, 2023, pp. 403–415.

[11] A. Stumpf and O. von Stryk, "A universal footstep planning methodology for continuous walking in challenging terrain applicable to different types of legged robots," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 420–10 427.

[12] S. McCrory, B. Mishra, R. Griffin, J. Pratt, and H. E. Sevil, "Bipedal navigation planning over rough terrain using traversability models," in *SoutheastCon 2023*. IEEE, 2023, pp. 89–95.

[13] D. Calvert, B. Mishra, S. McCrory, S. Bertrand, R. Griffin, and J. Pratt, "A fast, autonomous, bipedal walking behavior over rapid regions," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 24–31.

[14] B. Mishra, D. Calvert, B. Ortolano, M. Asselmeier, L. Fina, S. McCrory, H. E. Sevil, and R. Griffin, "Perception engine using a multi-sensor head to enable high-level humanoid robot behaviors," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9251–9257.

[15] Q. Tang, J. Liang, and F. Zhu, "A comparative review on multi-modal sensors fusion based on deep learning," *Signal Processing*, vol. 213, p. 109165, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168423002396

[16] T. Guan, R. Song, Z. Ye, and L. Zhang, "Vinet: Visual and inertial-based terrain classification and adaptive navigation over unknown terrain," in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 4106–4112.

[17] D. Stavens and S. Thrun, "A self-supervised terrain roughness estimator for off-road autonomous driving," *arXiv preprint arXiv:1206.6872*, 2012.

[18] V. Suryamurthy, Raghavan, Laurenzi, Tsagarakis, and Kanoulas, "Terrain segmentation and roughness estimation using rgb data: Path planning application on the centauro robot," in *Humanoids*, 2019, pp. 1–8.

[19] T. Guan, Z. He, R. Song, D. Manocha, and L. Zhang, "TNS: Terrain Traversability Mapping and Navigation System for Autonomous Excavators," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.

[20] S. M. Yoo, S. W. Hwang, D. H. Kim, and J. H. Park, "Biped robot walking on uneven terrain using impedance control and terrain recognition algorithm," in *Humanoids*, 2018.

[21] X. B. Peng, G. Berseth, and M. Van de Panne, "Terrain-adaptive locomotion skills using deep reinforcement learning," *ACM TOG*, vol. 35, no. 4, pp. 1–12, 2016.

[22] J. Xue, H. Zhang, and K. Dana, "Deep texture manifold for ground terrain recognition," in *CVPR*, 2018, pp. 558–567.

[23] S. Hosseinpoor, J. Torresen, M. Mantelli, D. Pitto, M. Kolberg, R. Maffei, and E. Prestes, "Traversability analysis by semantic terrain segmentation for mobile robots," in *CASE*. IEEE, 2021, pp. 1407–1413.

[24] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *RA-L*, 2022.

[25] M. Brandao, Y. M. Shiguematsu, K. Hashimoto, and A. Takanishi, "Material recognition cnns and hierarchical planning for biped robot locomotion on slippery terrain," in *Humanoids*. IEEE, 2016, pp. 81–88.

[26] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Learning and prediction of slip from visual information," *Journal of Field Robotics*, vol. 24, no. 3, pp. 205–231, 2007.

[27] B. Abbyasov, R. Lavrenov, A. Zakiev, K. Yakovlev, M. Svinin, and E. Magid, "Automatic tool for gazebo world construction: from a grayscale image to a 3d solid model," in *ICRA*. IEEE, 2020, pp. 7226–7232.

[28] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *CVPR*, 2018, pp. 2002–2011.

[29] X. Qi, Z. Liu, R. Liao, P. H. Torr, R. Urtasun, and J. Jia, "Geonet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation," *TPAMI*, 2020.

[30] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing," in *CVPR*, 2018, pp. 675–684.

[31] X. Wang, D. Fouhey, and A. Gupta, "Designing deep networks for surface normal estimation," in *CVPR*, 2015, pp. 539–547.

[32] A. Bansal, B. Russell, and A. Gupta, "Marr revisited: 2d-3d alignment via surface normal prediction," in *PCVPR*, 2016.

[33] Y. Man, X. Weng, X. Li, and K. Kitani, "Groundnet: Monocular ground plane normal estimation with geometric consistency," in *ACM MM*, 2019, pp. 2170–2178.

[34] M. Purri and K. Dana, "Teaching cameras to feel: Estimating tactile physical properties of surfaces from images," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*. Springer, 2020, pp. 1–20.

[35] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *International Conference on Computer Vision (ICCV) 2021*, 2021.

[36] J. DeGol, M. Golparvar-Fard, and D. Hoiem, "Geometry-informed material recognition," in *CVPR*, 2016.

[37] D. B. Gennery, "Traversability analysis and path planning for a planetary rover," *Autonomous Robots*, 1999.

[38] A. J. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," *SPME*, 2013.

[39] B. Sun and P. Mordohai, "Oriented point sampling for plane detection in unorganized point clouds," in *ICRA*, 2019.

[40] J. E. Lenssen, C. Osendorfer, and J. Masci, "Deep iterative surface normal estimation," in *CVPR*, 2020, pp. 11 247–11 256.

[41] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *CVPR*, 2017, pp. 6684–6692.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[43] X. Lyu, L. Liu, M. Wang, X. Kong, L. Liu, Y. Liu, X. Chen, and Y. Yuan, "Hr-depth: High resolution self-supervised monocular depth estimation," in *AAAI*, 2021.

[44] A. Sagar, "Monocular depth estimation using multi scale neural network and feature fusion," in *WACV*, 2022, pp. 656–662.

[45] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *TVCG*, vol. 9, no. 1, pp. 3–15, 2003.

[46] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Newcrfs: Neural window fully-connected crfs for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[47] I. Yun, H.-J. Lee, and C. E. Rhee, "Improving 360 monocular depth estimation via non-local dense prediction transformer and joint

supervised and self-supervised learning," in *AAAI*, 2022, pp. 3224–3233.

[48] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023. [Online]. Available: https://arxiv.org/abs/2302.12288

[49] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[50] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "Gmflow: Learning optical flow via global matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8121–8130.