2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

A Framework for Real-time Generation of Multi-directional Traversability Maps in Unstructured Environments

Tao Huang, Gang Wang, Hongliang Liu, Jun Luo, Lang Wu, Tao Zhu, Huayan Pu, Jun Luo, and Shuxin Wang

***Abstract*— In complex unstructured environments, accurate terrain traversability analysis is a fundamental requirement for the successful execution of any movements of ground robots, especially given that terrain traversability often exhibits anisotropy. However, the difficulty in obtaining multi-directional terrain labels hinders the emergence of end-to-end multi-directional traversability network. This paper introduces a framework for real-time multi-directional traversability maps (MTraMap) generation tailored for unstructured environments. It involves pre-training a uni-directional traversability classifier, termed UniTraT, through self-supervised learning using ground robot travel simulation. Furthermore, it employs Uni-directional to Multi-directional Traversability Distillation (UMTraDistill) to distill a multi-directional traversability network, termed MultiTCNN, which is capable of directly generating MTraMap. We evaluated both networks on our traversability dataset, achieving an 89% accuracy in terrain traversability classification with the UniTraT. Compared to UniTraT, the accuracy of the MultiTCNN distilled via UMTraDistill only decreases by 1.8%, and it can process 10 m × 10 m elevation map at a speed of 74 fps. Field robotics experiments were also conducted and showed that MultiTCNN can generate MTraMap of the surrounding 20 m × 20 m environment at a rate of 9.39 fps, with a slight reduction of 0.61 fps compared to the lidar data publishing rate, and the generated MTraMap can clearly delineate the multi-directional traversability of the surrounding environments.**

***Index Terms*—Foundations of automation, autonomous vehicle navigation, task planning.**

I. Introduction

Autonomous navigation of ground robots in unstructured environments is an extremely challenging task, especially in fields such as agriculture, mining, planetary exploration, and wilderness rescue, where terrain conditions often exhibit complexity and variability. Consequently, autonomous navigation of robots in unstructured environments has garnered significant attention from organizations like DARPA, NASA, and ESA, leading to many famous initiatives such as the Subterranean Challenge [1], Space Robotics Challenge [2], and ExoMars program [3]. Terrain traversability analysis serves as the foundation for robot navigation in such environments, representing the capability of robot to safely traverse various complex terrains [4].

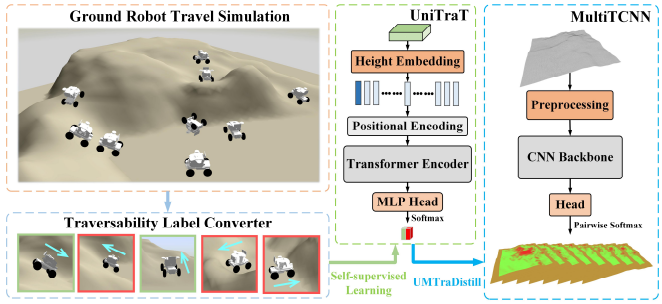


Fig. 1. Framework for MTraMap generation. The framework employs self-supervised learning to pre-train UniTraT from ground robot travel simulation and distills MultiTCNN using UMTraDistill.

However, current research predominantly focuses on how to determine the uni-directional traversability of terrain, often optimistically assuming uniform traversability in all directions [5]-[7], while neglecting potential variations in the robot's capability to traverse regions in different directions. Particularly in unstructured environments with steep slope and narrow path, terrain traversability often exhibits significant directional differences, known as anisotropy. For instance, on steep slope, ground robot may be able to descend but not ascend, as shown in Fig. 1. Multi-directional traversability maps (MTraMap) aid robots in considering the traversability of each location within the forward area from multiple directions when navigating complex terrains. This allows robots to find more optimal paths based on finer-grained traversability information. However, whether in real-world or simulated environments, obtaining traversability labels for a region in multiple directions is challenging, hindering the emergence of end-to-end multi-directional traversability network.

To acquire multi-directional terrain traversability, current methods typically involve training a uni-directional traversability classification or regression network, inputting the region into the network, and obtaining traversability for different directions through input rotation. Subsequently, to generate MTraMap of the surrounding environment, sliding window and rotating region on a global terrain map are applied to calculate traversability for different directions region by region [8] [9]. This method limits the ability to perform real-time traversability generation over large areas.

This work is supported in part by the National Natural Science Foundation of China (Nos. 62203072, U2013202, 61922053 and 62033001), the Natural Science Foundation of Chongqing, China (No. cstc2020jcyj-zdxmX0014), and the China Postdoctoral Science Foundation (No. 2022M710516). (Corresponding author: Gang Wang. E-mail: wangg@cqu.edu.cn).

T. Huang, G. Wang, H. L. Liu, J. Luo, L. Wu, H. Y. Pu, and J. Luo are with the State Key Laboratory of Mechanical Transmission for Advanced Equipments, College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China (e-mail: huangtao@stu.cqu.edu.cn; wangg@cqu.edu.cn; 20200701049g@cqu.edu.cn; 202207131098t@stu.cqu.edu.cn; wulangcqu@cqu.edu.cn; phygood_2001@shu.edu.cn; luoj@cqu.edu.cn).

T. Zhu is with the Key Laboratory of Optoelectronic Technology and Systems, Ministry of Education, Chongqing University, Chongqing 400044, China (e-mail: zhutao@cqu.edu.cn).

S. X. Wang is with the State Key Laboratory of Mechanical Transmission for Advanced Equipments, Chongqing University, Chongqing 400044, China (e-mail: shuxinw@tju.edu.cn).

This paper proposes a real-time MTraMap generation framework for ground robots in unstructured environments, as shown in Fig. 1. The framework employs self-supervised learning to pretrain a uni-directional traversability classifier and Uni-directional to Multi-directional Traversability Distillation (UMTraDistill) to distill a multi-directional traversability network without manual labeling. The distilled multi-directional traversability network can be used in real-time for MTraMap generation in unstructured environments. The main contributions of our work are as follows:

- A framework for real-time MTraMap generation in unstructured environments is proposed. The framework enables end-to-end MTraMap generation network to be obtained without manual labeling. The obtained network can achieve 87% accuracy on our traversability dataset and is able to process 10 m × 10 m elevation map at a speed of 74 fps. In addition, field robotics experiments were conducted and showed that MultiTCNN can generate MTraMap of the surrounding 20 m × 20 m environment at a rate of 9.39 fps, with a slight reduction of 0.61 fps compared to the lidar data publishing rate, and the generated MTraMap can clearly delineate the multi-directional traversability of the surrounding environments.
- A novel comprehensive definition of traversability map is provided, which takes into account both safety and directionality aspects for ground robots, guiding them to explore safely in unstructured environments where specific directional access is allowed.
- A transformer-based uni-directional traversability classification network is proposed, which introduces a self-attention mechanism to learn the relationship between ground robot's uni-directional traversability and the heights at various positions in terrain elevation patch. Additionally, a gradient encoding module is introduced to preprocess elevation data.
- A real-time multi-directional traversability network designed based on the required perceptual range for traversal is proposed, which can be utilized for multi-directional traversability classification of terrain elevation patch and is also capable of direct generation of MTraMap based on elevation map.

II. Related Work

Accurately analyzing terrain traversability is crucial for autonomous navigation of ground robots in unstructured environments. Existing terrain traversability methods can be broadly categorized into two main classes: traditional methods and deep learning methods. Traditional methods often rely on simple features or rules, such as roughness, slope, and step height [10], [11] extracted from point cloud or elevation map, or visual features like color, texture, and SURF [12]-[14] extracted from image, to analyze terrain traversability. In contrast, deep learning methods leverage neural networks to automatically learn deep features in terrain data and associate them with traversability for prediction.

Data-driven deep learning methods depend on extensive data support. Over the past decade, numerous terrain semantic datasets [15]-[17] for research on unstructured terrain traversability have been released. These datasets are manually labeled, but they do not encompass the actual traversability capabilities of robots. Accurately labeling the actual traversability capabilities of robots is challenging. Therefore, in recent years, many methods have employed self-supervised learning to directly learn terrain traversability from the interaction between robot and terrain. Some research involves real-world experiments where robots interact with terrain, associating their internal sensor data (e.g., data from odometry [18], IMU [19], force-torque sensor [20], or acoustic sensor [21]) with external sensor data to learn a terrain classifier for external sensor data.

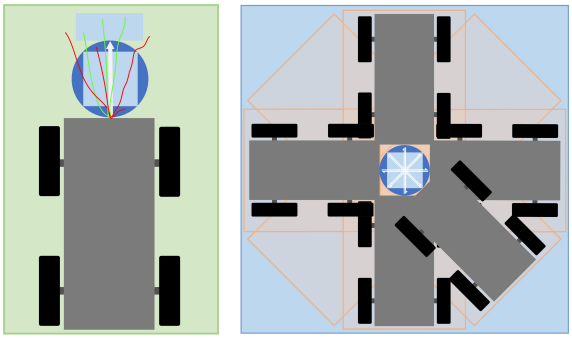
Considering that traversability methods tend to make overconfident predictions for out-of-distribution samples and it is challenging to collect negative sample data in real-world environments (due to the risk of robot damage in hazardous areas), some research has attempted to learn robot traversability from simulated environments. They place robots in various terrains for experimentation and learn a uni-directional terrain classifier [8], [22], [23] or a uni-directional regressor for traversability-related metrics [9] from the dynamic feedback of robot. Subsequently, different directions of traversability are obtained by rotating input regions and feeding them into the network. A sliding window approach is applied to calculate traversability for each region individually on a global terrain map, resulting in the generation of a global traversability map. However, calculating multi-directional traversability on a large area on a region-by-region basis is difficult to accomplish in real-time. Generating MTraMap in real-time is still an area that has not been fully explored.

III. Framework for MTraMap Generation

In this section, we propose a MTraMap generation framework for unstructured environment. This framework employs self-supervised learning through ground robot travel simulation for a uni-directional traversability classifier and utilizes UMTraDistill to distill a multi-directional traversability network for generating MTraMap. The following content will provide a detailed description of our framework.

A. Definition of Traversability Map

The definition of the Directional Traversability Map (DTraMap), is provided in this section, which represents the traversability of ground robot through terrain in a specific direction. For each grid cell in the DTraMap, its traversability is defined as the ability of the ground robot's front-end center point to safely traverse from one end of the grid cell to the other end in a specific direction while not exhibiting excessive lateral displacement. Since it is not convenient to represent the traversability in multi-direction using rectangular grid cell, the traversability of its circumcircle is used as an approximate substitute for the traversability of the corresponding grid cell. The traversability definition of the circumcircle is consistent with that of the rectangular grid cell. It is particularly noteworthy that, due to the fact that the circumcircle of the grid cell encompasses the grid cell, when the circular area is traversable in a certain direction, the corresponding grid cell in that direction is also traversable, whereas the opposite may not hold true, as shown in Fig. 2 (a). This is a safe approximation substitute. The MTraMap is composed of the DTraMap from different directions and is used to represent the traversability of all grid cells for terrain in various directions.



(a) (b)

Fig. 2. Uni-directional and multi-directional traversability of a grid cell.

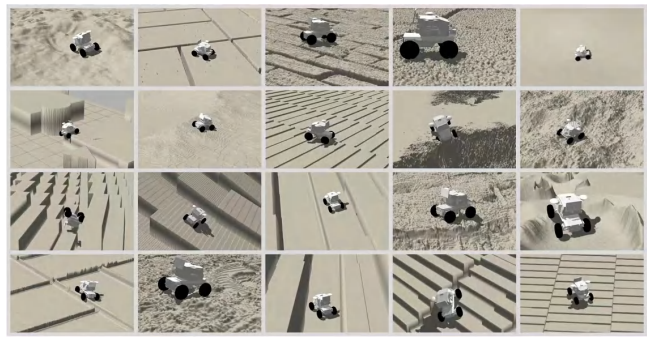


Fig. 3. Travel simulation of ground robot. The robot is placed in various terrains and positions and performs linear movement.

The terrain area covered by the robot as it traverses from one end of the circular area to the other is referred to as the required perception range for uni-directional traversal. The corresponding uni-directional elevation patch is taken as input for the uni-directional traversability network, as shown in the green region of Fig. 2 (a). Furthermore, due to variations in the robot's starting point and traversed area in different directions, the required perceptual range for traversal also differs in various directions, as shown in the orange region of Fig. 2 (b). The perceptual range required for multi-directional traversal is the union of these individual uni-directional perceptual ranges, with the corresponding multi-directional elevation patch taken as input for the multi-directional traversability network, as shown in the blue region of Fig. 2 (b).

B. *Uni-directional Traversability from Simulation*

Obtaining uni-directional traversability from simulation involves ground robot travel simulation and traversability label converter. Ground robot travel simulation enables the interaction of robot with terrain, thus providing motion trajectories and pose data for robot in different terrains. Meanwhile, the traversability label converter transforms the robot's trajectory and pose data into uni-directional traversability labels by integrating safety and traversability rules, thus generating a dataset of uni-directional traversability.

1) Ground Robot Travel Simulation: The travel simulation of ground robot in terrain scenarios can be divided into two steps. Firstly, the robot will be randomly placed at different locations within the terrain environment and assigned a random orientation, after which it will be required to remain stationary for a period of time. Subsequently, the robot will be guided to move at a constant speed v, within the terrain environment. During these two processes, real-time tracking of the robot's trajectory and pose data will be carried out to obtain static placement data and dynamic movement data, respectively. The scenarios and processes of the simulation are shown in Fig. 3.

2) Traversability Label Converter: The traversability label converter transforms the collected data into traversability labels corresponding to the terrain areas where the robot is intended to move. Firstly, static placement data is analyzed. When the robot is statically placed, it is essential to ensure that the robot does not roll, tip over, or experience significant displacement in the terrain. If the absolute value of the robot's Roll angle or Pitch angle exceeds the threshold values $thre_{\text{roll}}$ and $thre_{\text{pitch}}$, or if the robot's displacement after placement exceeds $thre_{\text{dist}}$, the robot placement area is considered statically unsafe, and the area in front of it is marked as non-traversable. Conversely, if the above conditions are not met, the area is classified as statically safe. Subsequently, an evaluation is performed on the dynamic movement data corresponding to statically safe areas. A region is considered traversable only when the robot can traverse it within the specified time while maintaining safety (no rolling or tipping over). During its movement, if the absolute value of the robot's Roll angle and Pitch angle are both below the threshold values $thre_{\text{roll}}$ and $thre_{\text{pitch}}$, and it traverses the area within time T while having lateral displacement less than the threshold value $thre_{\text{lateral}}$, the area is considered traversable. Otherwise, it is deemed non-traversable.

C. *Network Architecture for Traversability*

1) Uni-directional Traversability Transformer (UniTraT): UniTraT is employed within the uni-directional traversability network. UniTraT primarily consists of four key components: height embedding, positional encoding, transformer encoder [24], and MLP head, as shown in Fig. 4. This network is designed for uni-directional traversability classification of terrain elevation patch, determining whether the corresponding grid cell is uni-directionally traversable.

In the height embedding, the input terrain elevation patch is first processed using gradient encoding module, where the elevation at each position, except the edges, is encoded into gradients in eight adjacent directions. Subsequently, a gradient normalization operation is applied by subtracting the gradient mean μ_{grad} and dividing by the gradient standard deviation σ_{grad}, resulting in gradient sequences. Finally, a trainable linear projection is employed to map the 8-dimensional gradient sequences into a 128-dimensional embedded sequences.

Following this, a learnable traversability token is added to the embedded sequences, followed by positional encoding to preserve positional information within the embedded sequence in the terrain. Subsequently, the sequence with positional encoding is input into the transformer encoder. The transformer encoder comprises three transformer encoding layers. Each transformer encoding layer encompasses multi-head self-attention, layer normalization, and multilayer perceptron (MLP) operations. In these layers, the self-attention mechanism enable interactions between gradients at each terrain position (or traversability token) with gradients at all other positions, capturing the relationship

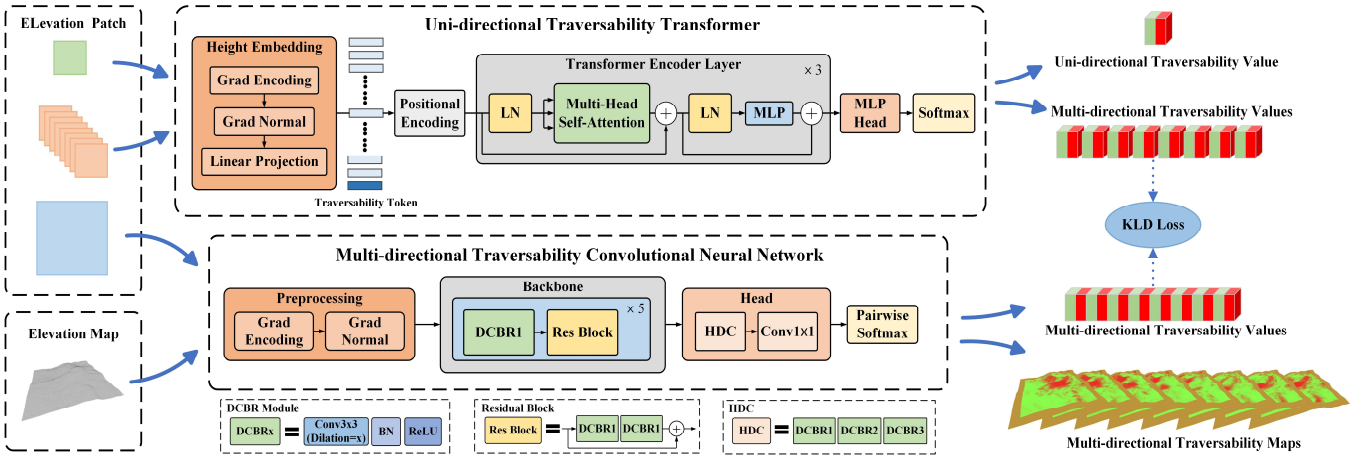


Fig. 4. Traversability networks and uni-directional to multi-directional traversability distillation. DCBRx denotes the dilated convolution layer, which is converted to the conventional convolution layer when the dilation rate is one.

between the robot's traversability within elevation patch and terrain gradients.

Ultimately, the traversability token vector encoded by the transformer encoder is fed into the head with MLP for traversability prediction. The final traversability category probability output is obtained through the Softmax function.

2) Multi-directional Traversability Convolutional Neural Network (MultiTCNN): The multi-directional traversability network utilizes the MultiTCNN. This network comprises three components: height preprocessing, backbone, and head, as shown in Fig. 4. MultiTCNN can be used for multi-directional traversability classification of terrain elevation patch and for generating MTraMap of terrain elevation map.

In the height preprocessing, similar to UniTraT, terrain elevation map is fed into the gradient encoding module for encoding, followed by the gradient normalization, resulting in an 8-channel gradient maps. The backbone employs five convolutional layers and residual blocks [25] to process these gradient maps, extracting high-dimensional features of the terrain, and generating feature maps. The head utilizes Hybrid Dilated Convolution (HDC) [26] to process the feature maps, simultaneously expanding the receptive field of network [27]. Subsequently, a 1×1 convolutional layer is used to predict multi-directional traversability, resulting in 16 channels of prediction maps. Finally, Pairwise Softmax is applied individually to each pair of channels in each direction, yielding eight directional traversability and non-traversability probability maps.

It is worth emphasizing that the overarching design objective of the entire network is to simultaneously cater to both multi-directional traversability classification of terrain elevation patch and MTraMap generation for elevation map. To achieve this, the receptive field of the network has been carefully crafted to match the required perception range for multi-directional traversal, with the added feature that no padding operation is employed across all convolutional layers. When elevation patch is inputted into the network, the output consists of 16 values, which represent the traversability and non-traversability probabilities of the corresponding grid cell in eight different directions. When elevation map is used as input, the network generates 16 maps, which represent the traversability and non-traversability probability maps in eight different directions across the elevation map.

D. Uni-directional to Multi-directional Traversability Distillation

In order to obtain the MultiTCNN network for generating MTraMap, UMTraDistill is employed. MultiTCNN serves as the student model, while UniTraT serves as the teacher model. The UniTraT model is utilized to perform uni-directional to multi-directional traversability distillation on the MultiTCNN.

As shown in Fig. 4, firstly, the uni-directional labels transformed by the traversability label converter, along with their corresponding uni-directional elevation patches (the green region), are utilized for the pre-training of the teacher network. Subsequently, the trained teacher model is employed to predict uni-directional elevation patches from the same grid in eight adjacent directions (the orange region), resulting in eight-directional traversability and non-traversability probabilities. These probabilities are then used as soft labels for the student model and are employed in conjunction with their corresponding multi-directional elevation patches (the blue region) for the distillation of the multi-directional traversability network.

During the pre-training of the teacher model, the cross-entropy loss [28] is employed. Meanwhile, in the uni-directional to multi-directional traversability distillation process, the Kullback-Leibler divergence (KLD) loss [29] is employed to quantify the differences in the probability distribution of traversability between the teacher network and the student network in eight directions.

IV. Experimental Results

A. Simulation and Experimental Setup

1) Simulation Setup: Physical simulations were conducted using the Gazebo simulator and the ODE physics engine [30]. Ground robot simulations were performed on 299 simulated terrains, encompassing elements of unstructured environments such as stair, ramp, pothole, rugged terrain, desert, and mountain, as shown in Fig. 3. Each terrain is sized at 21 m × 21 m, with heights ranging from 0 to 2 m. The dynamics of the Hunter SE robot (of size 82 cm × 69 cm × 68 cm) were modeled in gazebo to match the real robot, its motion was controlled at a velocity of 0.2 m/s, and data was collected at a rate of approximately 50 Hz.

2) Network Training: In all the experiments described below, the elevation map has a grid resolution of 4 cm × 4 cm, which is fine enough to represent the terrain. In the traversability label converter, $thre_{roll} = 45^{\circ}$, $thre_{pitch} = 45^{\circ}$, $thre_{dist}$ = 50 cm, T = 0.8 s, and $thre_{lateral}$ = 6 cm. The uni-directional traversability dataset transformed by the traversability label converter includes 1098 k training samples, 236 k validation samples, and 236 k testing samples. In gradient normalization, $\mu_{grad} = 0$ cm and $\sigma_{grad} = 6.41$ cm. The inputs to the networks are elevation patches that have a larger perceptual range than required by their respective traversability. UniTraT has an input size of 27 grids × 27 grids, while MultiTCNN has an input size of 45 grids × 45 grids. Adam [31] optimizer with a momentum of 0.9 was employed for all networks. UniTraT was trained with a batch size of 540, 80 epochs, utilizing a cosine learning rate decay [32] starting at 0.0005 and a minimum learning rate of 0.000005. MultiTCNN was trained with a batch size of 120, 96 epochs, employing a cosine learning rate decay starting at 0.001 and a minimum learning rate of 0.00001. Training was conducted on four NVIDIA Tesla P100 GPUs with 16GB memory.

B. Performance of Traversability Networks

1) Performance Metrics of UniTraT and MultiTCNN: The classification accuracy and inference speed of the UniTraT and MultiTCNN networks were evaluated. Since the dataset contains only uni-directional traversability labels, the performance metrics for the multi-directional traversability network were validated individually for each uni-directional traversability accuracy. The speed of MTraMap generation for uni-directional traversability network is the speed of calculating the multi-directional traversability values region-by-region at the elevation map using the sliding window and rotating region methods.

As shown in Table I, UniTraT network achieved an accuracy of 89% on both the validation and test sets, indicating that UniTraT, which utilizes the gradient encoding and the transformer encoder, can effectively learn the relationship between terrain elevation patch and traversability and correctly classify traversability. In comparison, MultiTCNN exhibited a slight decrease in accuracy of only 1.8%, indicating that knowledge of traversability learned by the uni-directional network can be effectively transferred to the multi-directional network through UMTraDistill.

Furthermore, the end-to-end MultiTCNN demonstrated excellent real-time performance, with an inference speed of 74 fps for 10 m × 10 m elevation map and 33 fps for 15 m × 15 m elevation map. This performance significantly surpasses the region-by-region computation speed of UniTraT, which is less than 0.001 fps. Consequently, MultiTCNN is highly suitable for real-time MTraMap generation.

TABLE I
CLASSIFICATION ACCURACY AND INFERENCE SPEED OF THE NETWORKS. THE SPEEDS WERE MEASURED ON NVIDIA JETSON AGX ORIN.

Network	Acc_{val}	Acc_{test}	FPS_{cls}	FPS_{10m}	FPS_{15m}
UniTraT	89.24%	89.29%	180	<0.001	
MultiTCNN	87.42%	87.53%	199	74	33

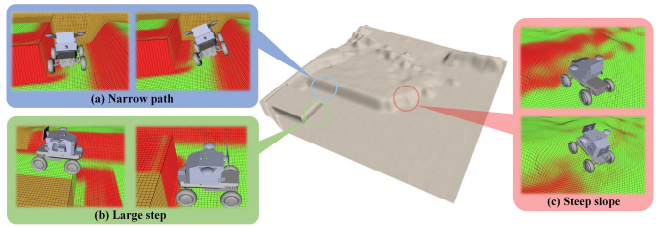


Fig. 5. Unstructured terrain and robot traversability demonstration on steep slope, narrow path, and large step specific areas.

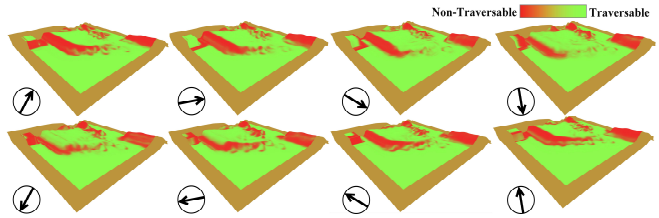


Fig. 6. Traversability maps in eight directions. Redder areas are harder to traverse, greener areas are easier. The brown areas mark regions of unknown traversability due to incomplete determination at raster edges.

2) Evaluation of MultiTCNN: MultiTCNN was validated on unstructured terrains in simulation, and the MTraMap, as shown in Fig. 5 and Fig. 6, reveal distinct traversability characteristics in different directions. On flat terrains and gentle slopes, all directions are considered traversable. However, when encountering large obstacles, all directions are deemed non-traversable. For steep slope, narrow path, and large step, only certain directions are considered traversable. Following are some typical case analyses.

Traversability of steep slope. Steep slope is regarded as non-traversable when moving uphill and traversable when moving downhill, as shown in Fig. 5 (c) and Fig. 6. This aligns with real-world conditions.

Traversability of narrow path. In the case of narrow path, only the direction parallel to the narrow path allows traversal, as shown in Fig. 5 (a) and Fig. 6. Furthermore, placing the robot at the extreme left of the traversable area within the narrow path positions it precisely at the left edge. When placing the robot at the extreme right of the traversable area within the narrow path, despite its tilted pose, it can move straight through the area, confirming the accuracy of the traversability calculation.

Traversability of large step. At the edge of large step, it is considered an obstacle when moving in the direction from below the step going upward, and it is deemed non-traversable. Conversely, when moving in the opposite direction, it is considered a cliff, and the transition area is considered traversable. Considering that the grid represents whether the robot's front-end center point can safely traversal, in this case,

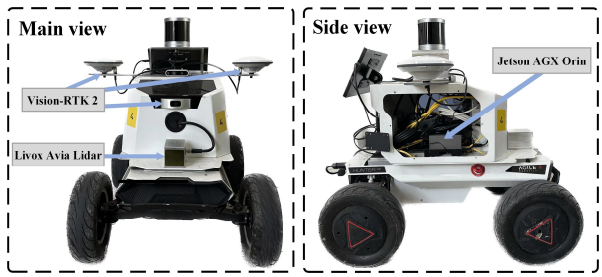


Fig. 7. Hunter SE robot used for experiment. Hunter SE is equipped with a Livox Avia lidar, a Vision-RTK 2 positioning sensor to provide point cloud and pose data for the elevation mapping method, and an Nvidia Jetson AGX Orin mainboard to run all methods.

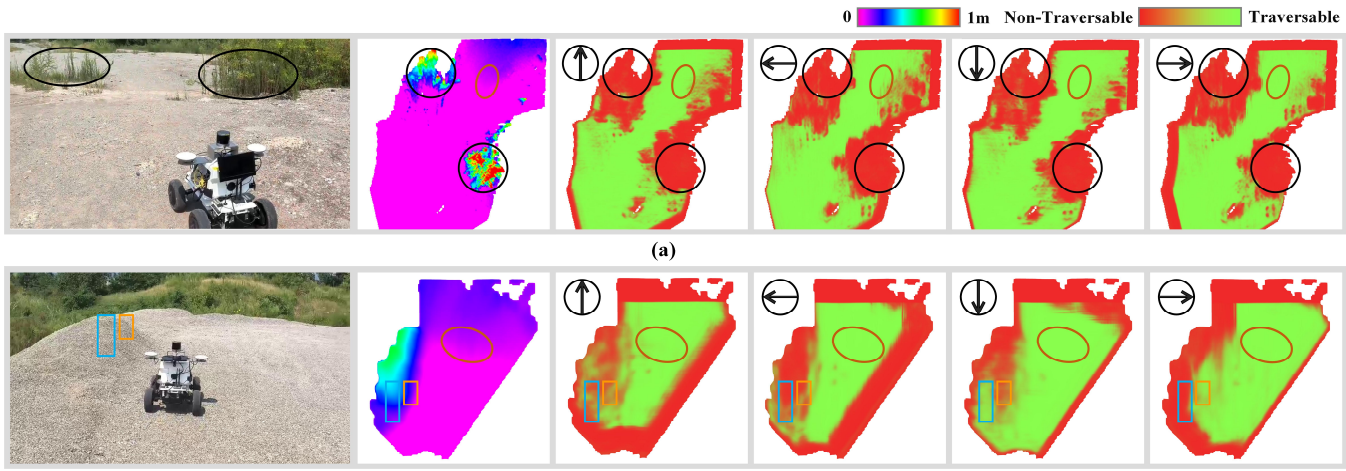


Fig. 8. MTraMap generated in real-world environments. From left to right: real terrain environment, elevation map, traversability maps for four directions.

the robot's front center point can indeed safely reach the edge of the step without falling, as shown in Fig. 5 (b).

C. Field Robotics Experiments

1) Effectiveness of MTraMap: We conducted experiments on MultiTCNN in a rugged quarry using the Hunter SE robot. The Hunter SE is equipped with a Livox Avia lidar, Vision-RTK 2 positioning sensor, and Nvidia Jetson AGX Orin mainboard, as shown in Fig. 7. A GPU-based elevation mapping method [33] was employed to provide elevation map inputs for MultiTCNN. This method utilizes point cloud and robot pose data, leveraging the GPU for real-time and efficient elevation map construction.

As shown in Fig. 8, despite the noticeable error in the elevation map constructed in challenging terrains, proposed method can still clearly delineate areas that are traversable. Slightly rough terrain is considered traversable, as shown in the brown regions of Fig. 8. Meanwhile, thick vegetation is deemed non-traversable, as shown in the black regions of Fig. 8 (a). For steep slopes, the generated traversability map exhibits excellent multi-directionality, as shown in Fig. 8 (b). In the blue region, the upward direction is deemed non-traversable, while the downward direction is traversable. In the orange region, the left direction is non-traversable, while the right direction is traversable. The robot can easily descend steep slopes, but ascending them is more challenging, which aligns with real-world conditions. Furthermore, both upward and downward directions in the orange region are non-traversable because the robot may risk tipping over at the edges of steep slopes.

2) Speed of MTraMap Generation: We also tested the generation speed of MTraMap in the quarry, as shown in Table Ⅱ. The MultiTCNN exhibits outstanding real-time performance, capable of generating MTraMap at a rate nearly matching the lidar data publishing rate. It generates MTraMap for the surrounding 10 m × 10 m environment at a rate of 9.93 fps, with only a minimal decrease of 0.07 fps compared to the lidar data publishing rate and a decrease of 0.01 fps compared to the elevation map construction rate. At a rate of 9.39 fps, it generates MTraMap for the surrounding 20 m × 20 m environment, with a slight reduction of 0.61 fps compared to the lidar data publishing rate and a decrease of 0.48 fps compared to the elevation map construction rate.

TABLE II
GENERATION SPEED OF MTRAMAP. FPS_{sensor} IS THE PUBLISHING RATE OF LIDAR DATA, FPS_{ele} IS THE SPEED OF ELEVATION MAP CONSTRUCTION, AND FPS_{tra} IS THE SPEED OF MTRAMAP GENERATION.

Elevation map size	FPS_{sensor}	FPS_{ele}	FPS_{tra}
10m×10m	10	9.94	**9.93**
15m×15m		9.91	**9.90**
20m×20m		9.87	**9.39**

V. CONCLUSION

In this work, we propose a comprehensive framework for MTraMap generation tailored for unstructured environments, which distills MultiTCNN to generate traversability maps for robots in multiple directions in real-time. To achieve this goal, we pre-train UniTraT in a self-supervised manner through ground robot travel simulation. Subsequently, we employ UMTraDistill to distill MultiTCNN. We evaluate both networks on our traversability dataset, achieving an 89% accuracy in terrain traversability classification with the UniTraT. Compared to UniTraT, the accuracy of the MultiTCNN distilled via UMTraDistill only decreases by 1.8%, and it can process 10 m × 10 m elevation map at a speed of 74 fps. Furthermore, We validate MultiTCNN through field robotics experiments, and the results show that MultiTCNN can generate MTraMap of the surrounding 20 m × 20 m environment at a rate of 9.39 fps, with a slight reduction of 0.61 fps compared to the lidar data publishing rate, and the generated MTraMap can clearly delineate the multi-directional traversability of the surrounding environments. This demonstrates the practicality of proposed frame for real-time generation of MTraMap in unstructured environments.

Future work includes conducting extensive comparative experiments, optimizing elevation map construction, and improving Hybrid A* [34] planning method to leverage MTraMap for autonomous navigation in open unstructured environments.

REFERENCES

[1] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, ... and K. Alexis, "CERBERUS in the DARPA Subterranean Challenge," *Sci. Rob.*, vol. 7, no. 66, pp. eabp9742, 2022.

[2] R. Sachdeva, R. Hammond, J. Bockman, A. Arthur, B. Smart, D. Craggs, ... and I. Reid, "Autonomy and perception for space mining," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4087-4093.

[3] A. M. Barrett, M. R. Balme, M. Woods, S. Karachalios, D. Petrocelli, L. Joudrier, and E. Sefton-Nash, "NOAH-H, a deep-learning, terrain classification system for Mars: Results for the ExoMars Rover candidate landing sites," *ICARUS*, vol. 371, pp. 114701, 2022.

[4] P. Borges, T. Peynot, S. Liang, B. Arain, M. Wildie, M. Minareci, ... and P. Corke, "A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges," *Field Robot.*, vol. 2, no. 1, pp. 1567-1627, 2022.

[5] M. G. Castro, S. Triest, W. Wang, J. M. Gregory, F. Sanchez, J. G. Rogers, and S. Scherer, "How does it feel? self-supervised costmap learning for off-road vehicle traversability," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 931-938.

[6] R. Schmid, D. Atha, F. Schöller, S. Dey, S. Fakoorian, K. Otsu, ... and A. A. Agha-mohammadi, "Self-supervised traversability prediction by learning to reconstruct safe terrain," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12419-12425.

[7] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. Velasquez, V. A. Higuti, J. Rogers, ... and G. Chowdhary, "Wayfast: Navigation with predictive traversability in the field," *IEEE Rob. Autom. Lett.*, vol. 7, no. 4, pp. 10651-10658, 2022.

[8] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, "Learning ground traversability from simulations," *IEEE Rob. Autom. Lett.*, vol. 3, no. 3, pp. 1695-1702, 2018.

[9] E. Wallin, V. Wiberg, F. Vesterlund, J. Holmgren, H. J. Persson, and M. Servin, "Learning multiobjective rough terrain traversability," *J. Terramech.*, vol. 102, pp. 17-26, 2022.

[10] T. Overbye, and S. Saripalli, "G-VOM: A GPU Accelerated Voxel Off-Road Mapping System," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 1480-1486.

[11] Z. Jian, Z. Lu, X. Zhou, B. Lan, A. Xiao, X. Wang, and B. Liang, "Putn: A plane-fitting based uneven terrain navigation framework," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 7160-7166.

[12] P. Filitchkin, and K. Byl, "Feature-based terrain classification for littledog," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1387-1392.

[13] Y. Zou, W. Chen, L. Xie, and X. Wu, "Comparison of different approaches to visual terrain classification for outdoor mobile robots," *Pattern Recognit. Lett.*, vol. 38, pp. 54-62, 2014.

[14] H. Pu, J. Luo, G. Wang, T. Huang, H. Liu and J. Luo, "Visual SLAM Integration With Semantic Segmentation and Deep Learning: A Review," *IEEE Sens. J.*, vol. 23, no. 19, pp. 22119-22138, 2023.

[15] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," in *2021 IEEE international conference on robotics and automation (ICRA)*, 2021, pp. 1110-1116.

[16] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *Int. J. Comput. Vision*, vol. 127, pp. 302-321, 2019.

[17] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5000-5007.

[18] M. Nava, J. Guzzi, R. O. Chavez-Garcia, L. M. Gambardella, and A. Giusti, "Learning long-range perception using self-supervision from short-range sensors and odometry," *IEEE Rob. Autom. Lett.*, vol. 4, no. 2, pp. 1279-1286, 2019.

[19] K. Otsu, M. Ono, T. J. Fuchs, I. Baldwin, and T. Kubota, "Autonomous terrain classification with co-and self-training approach," *IEEE Rob. Autom. Lett.*, vol. 1, no. 2, pp. 814-819, 2016.

[20] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Rob. Autom. Lett.*, vol. 4, no. 2, pp. 1509-1516, 2019.

[21] J. Zürn, W. Burgard, and A. Valada, "Self-supervised visual terrain classification from unsupervised acoustic feature learning," *IEEE Trans. Rob.*, vol. 37, no. 2, pp. 466-481, 2020.

[22] J. Seo, T. Kim, K. Kwak, J. Min, and I. Shim, "Scate: A scalable framework for self-supervised traversability estimation in unstructured environments," *IEEE Rob. Autom. Lett.*, vol. 8, no. 2, pp. 888-895, 2023.

[23] 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)*, 2022, pp. 5722-5729.

[24] Q. Qiu, H. Gao, W. Hua, G. Huang, and X. He, "Priorlane: A prior knowledge enhanced lane detection approach based on transformer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5618-5624.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[26] W. Jiang, M. Liu, Y. Peng, L. Wu, and Y. Wang, "HDCB-Net: A neural network with the hybrid dilated convolution for pixel-level crack detection on concrete bridges," *IEEE Trans. Ind. Inf.*, vol. 17, no. 8, pp. 5485-5494, 2020.

[27] J. Bruton, and H. Wang, "Translated Skip Connections-Expanding the Receptive Fields of Fully Convolutional Neural Networks," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 631-635.

[28] H. Masnavi, J. Shrestha, M. Mishra, P. B. Sujit, K. Kruusamäe, and A. K. Singh, "Visibility-aware navigation with batch projection augmented cross-entropy method over a learned occlusion cost," *IEEE Rob. Autom. Lett.*, vol. 7, no. 4, pp. 9366-9373, 2022.

[29] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8479-8488.

[30] 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7226-7232.

[31] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Rob. Autom. Lett.*, vol. 7, no. 2, pp. 4606-4613, 2022.

[32] W. Shu, K. Cai, and N. N. Xiong, "A short-term traffic flow prediction model based on an improved gate recurrent unit neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16654-16665, 2021.

[33] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using gpu," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2273-2280.

[34] M. Thoresen, N. H. Nielsen, K. Mathiassen, and K. Y. Pettersen, "Path planning for UGVs based on traversability hybrid A," *IEEE Rob. Autom. Lett.*, vol. 6, no. 2, pp. 1216-1223, 2021.