2022 8th International Conference on Automation, Robotics and Applications (ICARA), Prague,
18 February - 20 March 2022
DOI:10.1109/ICARA55094.2022.9738557

Hybrid Terrain Traversability Analysis in Off-road Environments

Tiga Ho Yin Leung
School of Aerospace
Cranfield University
Cranfield, United Kingdom
Tiga.leung@cranfield.ac.uk

Dmitry Ignatyev
School of Aerospace
Cranfield University
Cranfield, United Kingdom
D.Ignatyev@cranfield.ac.uk

Argyrios Zolotas
School of Aerospace
Cranfield University
Cranfield, United Kingdom
A.Zolotas@cranfield.ac.uk

***Abstract*—There is a significant growth in autonomy level in off-road ground vehicles. However, unknown off-road environments are often challenging due to their unstructured and rough nature. To find a path that the robot can move smoothly to its destination, it needs to analyse the surrounding terrain. In this paper, we present a hybrid terrain traversability analysis framework. Semantic segmentation is implemented to understand different types of the terrain surrounding the robot; meanwhile geometrical properties of the terrain are assessed with the aid of a probabilistic terrain estimation. The framework represents the traversability analysis on a robot-centric cost map, which is available to the path planners. We evaluated the proposed framework with synchronised sensor data captured while driving the robot in real off-road environments. This thorough terrain traversability analysis will be crucial for autonomous navigation systems in off-road environments.**

***Index Terms*—terrain traversability analysis, off-road environments, elevation mapping, semantic segmentation, cost map**

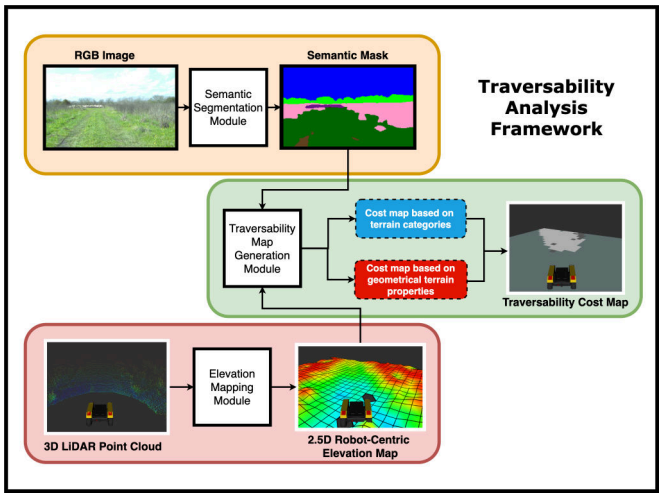

Fig. 1. Overview of the terrain traversability analysis framework.

I. Introduction

Autonomous navigation in an off-road environment is full of potential. It offers a wide range of possibilities and applications such as forestry monitoring, mining, planetary exploring, searching, and rescuing [1]. However, unlike urban and structured environments, off-road environments lack a viable path or helpful landmark supporting autonomous navigation [2]. Moreover, the driving surface is uneven and not consistent. Driving on different types of terrain exhibits different behaviour. Operating in these unstructured off-road environments, the agent needs to analyze the terrain traversability and plan an optimal and terrain-aware path according to it [1][3]. Traversability analysis is defined as estimating the difficulty of driving through terrain for an off-road ground vehicle[2].

In this paper, we are interested in finding out how the robot can assess and understand its surrounding terrain in the off-road environment. Understanding means knowing what types of terrain is surrounding the robot. Assessing refers to estimating a cost if the robot traverses to that specific area.

Recent researches [4]-[6] proposed a geometrical approach for surveying the traversability of the terrain forehead. Wermelinger [4] and Fankhauser[5] generated a 2.5D robot-centric elevation map from the LIDAR point cloud information and robot odometry. Later, the traversability score was computed from the terrain's slope, roughness and step height which was derived from the height information stored in the elevation map. However, since the geometrical approaches neglect the semantic information of the surrounding environment, it cannot classify different types of the terrain ahead of the robot. For example, the robot can easier move across a patch of grass instead of a patch of sand even though the geometrical properties of both are similar. A pure geometrical approach is overlooking this vital navigation information.

Meanwhile, some works [7]–[10] suggested using a pure vision approach that segmentizes the images captured from the front camera based on its terrain types. Semantic information of the surrounding environment is attained. However, only relying on this information is not enough to analyze the terrain traversability. The uneven driving surface in the off-road environment will also cause difficulties for the robot to travel to some areas.

This paper contributes a novel and hybrid framework for traversability analysis targeted at the off-road environment. Semantic segmentation helps to identify different types of the terrain surrounding the robot, such that the robot can optimize its path using more appropriate terrains, such as concrete or grass. Meanwhile assessing the geometrical properties of the

surrounding terrain with the aid of an elevation map, allows us to take the physical capabilities of the robotic platform into account for the traversability analysis. The framework presents the traversability information on a 2D robot-centric cost map, which is accessible for path planners. We believe this well-rounded analysis of the surroundings will help to improve mission performance of an off-road robot platform. The demo video can be found at: https://youtu.be/lgh_6czhnR0.

The remainder of this paper is organized as follows. Section II compares our proposed framework with some previous works. Section III describes our hybrid approach to traversability cost map generation. Section IV presents our experiment settings and the experimental results. In section V, we draw a conclusion and discuss the future work.

II. Related Work

Recently, Guastella [2] clearly defined terrain traversability analysis and surveyed its latest approaches in unstructured environments. Terrain traversability analysis estimates the difficulty for a ground vehicle to drive through a specific patch of terrain. Comparing to obstacles avoidance, an obstacle may not necessarily exists in terrain traversability analysis. It is helping the robot to make a trade-off between 2 different unobstructed paths. Fankhauser [5] proposed a novel approach to real-time elevation mapping which addresses the problem of localization drift for the legged robots. The grid-based elevation map is a probabilistic estimation of the terrain. Each grid cell includes an estimated height and its upper and lower confidence bounds. Wermelinger [4] made use of the elevation map and presented a framework for planning a safe and efficient path for the legged robot in rough and unstructured terrain. Geometrical terrain properties are derived from the height information stored in each grid cell. Subsequently, a traversability score is computed from these properties and stored in a traversability cost map.

In this work, we built upon the elevation mapping method[5] and borrowed the idea for calculating the traversability cost from the terrain geometrical properties [4]. However, as opposite to the above mentioned papers, we are considering not only geometrical properties of the terrain but also the types of the terrain. The traversability analysis framework is enhanced by adding a vision-based semantic segmentation of the surrounding terrain. A novel approach is proposed to utilize both geometrical and semantic data. Thus, we can obtain the types of terrain ahead and present them on the traversability cost map as well. The robot can plan a path according to different types of terrain ahead of it; e.g. avoiding to move on top of the puddle or preferring to stay on the grass.

On the other hand, Rothrock [9] presented the first work using image semantic segmentation for traversability analysis. It was used to identify a feasible landing site for the Mars 2020 rover mission. Orbital or ground-acquired images were fed into a deep convolutional neural network (DCNN). It segmented the image according to its terrain types. Valada [12] proposed the UpNet architecture for semantic segmentation in forested environments. Multi-spectral and multimodal images are used for semantic segmentation. A DCNN fusion architecture is proposed to fuse various channels (namely RGB, Near-InfraRed, and depth images). The segmentation results show the fusion approach outperformed other approaches using either one of the modalities. These previous works have proven the promising of using semantic segmentation in analyzing terrain types. Similar to their approaches, we feed the captured images into a deep convolutional neural network, yielding a segmented semantic mask over the images which identify different types of the terrain ahead. Instead of simply presenting the semantic information on a 2D pixel plane, we re-projected the terrain information and represented it on a robot-centric traversability cost map that can be utilized by a path planning algorithm such as A* or Rapidly-Exploring Random Tree (RRT)[24].

Maturana [10] and Zhao [13] shared a similar approach to our work. While Maturana [10] only assigned a traversability reward "0.1" to the patches of terrain which were identified as grass and leaving others to "0". In contrast to this, we adopt a finer cost scheme that assigns different empirical traversability costs to different types of terrain, allowing the agent to survey its surrounding environment comprehensively. Zhao [13] also incorporated the elevation mapping approach proposed by [4] [5] into his work and introduced a deep neural network based semantic segmentation model to provide semantic labels of the images. Contrary to Zhao's work, we target operating in an unstructured, off-road environment instead of an urban environment with a well-constructed path. To this end, we introduce a hybrid approach for terrain traversability analysis. It analyses the traversability of the surrounding environment based on terrain types and geometrical properties. We believe the traversability cost estimated from the geometrical properties can compensate for the downside of the pure semantic segmentation-based approach.

III. Method

A. System Overview

The proposed framework assesses terrain traversability and represents them on a robot-centric cost map. It takes RGB image, LIDAR point cloud and robot motion information as inputs, and outputs a fused traversability cost map that is computed from both terrain types and geometrical properties. The overview of the framework is illustrated in Fig. 1. The cost map generated by the proposed framework can not only be visualised on screen but more importantly, they can be used by the path planning algorithm to plan a terrain-aware path.

The framework is divided into 3 main modules: Semantic Segmentation, Elevation Mapping and Traversability Map Generation. The semantic segmentation module outputs a colourized semantic mask indicating the location of different types of terrain on the 2D pixel plane. The elevation mapping module is introduced to generate a 2.5D robot-centric elevation map. Grids' positions on the map are all related to a map coordinate frame in the 3D world coordinate system. And each grid cell holds the height information of its corresponding position. Last but not least, the traversability map generation

module incorporates the terrain semantic information with the robot-centric cost map. Also, it surveys the surrounding terrain based on its geometrical properties derived from the height information. Subsequently, it fuses both traversability information into one and represents it on a cost map. These modules work asynchronously. The traversability map generation module is triggered by a timer. Whenever it is triggered, it requests the latest elevation map and semantic mask from the elevation mapping and semantic segmentation module. All the modules are implemented as a Python and C++ library with an interface to the Robotics Operating System (ROS).

B. Semantic Segmentation Module

The semantic segmentation module is the first element of the whole system. It assigns each pixel in the input image a label from a predefined set of terrain categories. It allows the robot to get a comprehensive understanding of the surrounding environment, such as where the traversable terrain (grass, concrete, dirt) or less (puddle, rocks) or even inaccessible terrain (trees, fences) are located.

It takes an RGB image with a resolution of 1920×1200 from the camera. Then, the image is rescaled into 640×400 to match the network's input size requirement. The trained neural network will then process the input image and output a colourised semantic mask that has the same resolution as the rescaled input image.

In this work, 2 different deep neural network architectures were tested and compared. The Gated-SCNN architecture proposed by Takikawa [14] is the first candidate and ERFNet proposed by Romera [11] is the second candidate architecture. Both of them were implemented in the PyTorch framework [15]. A detailed comparison between the 2 network architectures was achieved in this work. The results are shown in Section IV. ERFNet architecture outer-performed the Gated-SCNN architecture in real-time performance, while the accuracy advantage of Gated-SCNN was not notable. As a result, ERFNet was chosen to be the neural network architecture for the semantic segmentation module.

Both neural networks were trained in a supervised fashion with the RELLIS-3D dataset contributed by Jiang [16]. It is a multi-modalities off-road navigation dataset with synchronised raw sensor data and a large number of ground truth annotations. There are 18 classes of terrain presented in the dataset. We split the dataset into a training set of 3,302 images, a validation set of 983 images and a testing set of 1,672 images.

For training the ERFNet, the encoder was trained independently at first. Extra pooling layers and a fully connected layer were added to the encoder's last layer. Then the modified encoder network was trained with the ImageNet dataset [17]. Afterwards, these extra pooling layers and fully connected layers were removed. Thus, the weights of the encoder network were initialised by training with the ImageNet dataset. Next, the decoder network was attached to the bottom of the encoder network. The entire network was trained with the RELLIS-3D dataset. The model was trained to perform inference at 640×400 to reduce the computational cost, as to improve the real-time performance. Adam optimisation of stochastic gradient descent (SGD) was used. The training was performed with the momentum of 0.9, weight decay of 2×10^{-4}, batch size of 2, and learning rate started with 5×10^{-4} and was divided by a factor of 2 every time that the training error became stagnant. The network was trained for 150 epochs while the best performing weight was saved.

C. Elevation Mapping Module

This module is responsible for real-time terrain mapping. We have used the elevation mapping library developed by Fankhauser [5] to generate the 2.5D robot-centric elevation map. It takes the 3D point cloud from the LIDAR and robot motion information as inputs and generates the 2.5D robot-centric elevation map as an output. Each grid on the map holds an estimation of the height and its upper and lower confidence bounds at that specific location. The positions of the grids are related to a stable map coordinate frame in the 3D world coordinate system. Thus, it helps the subsequent module for projecting the semantic information from the 2D pixel space to the actual 3D space.

D. Traversability Map Generation Module

1) Cost map based on terrain types: Modern reliable path planning algorithms require a robot-centric cost map as a medium for it to understand the surrounding environment. The positions of each grid on the map are related to a 3D world coordinate frame. Representing the locations of different types of terrain on a pixel plane is not desired for the planning algorithms. This module projects the locations of different types of terrain from 2D pixel coordinates to the 3D world coordinates and represents it on a 2D robot-centric cost map. Thus, an empirical traversability cost can be assigned on each grid based on the terrain type it belongs to. This allows the path planning algorithm to plan an optimal, terrain-aware path by accessing the traversability analysis from the cost map.

Algorithm 1 Traversability Cost Assignation

for grid cell $i = 1, 2, \ldots, n$ in elevation map M **do**
  $(x_i, y_i, z_i) = GetGrid3DPosition(i)$
  $(x_i, y_i, z_i) = FilterMap(x_i, y_i, z_i)$
  $(u_i, v_i) = ProjectionTransform(x_i, y_i)$ (Eq.: 3&4)
  $colour = CheckRGBValue(u_i, v_i)$
  if $colour == Colour_{grass}$ **then**
    $t_t \leftarrow 0.1$
  else if $colour == Colour_{water}$ **then**
    $t_t \leftarrow 0.5$
  else if $colour == Colour_{bush}$ **then**
    $t_t \leftarrow 0.8$
  else if $\cdots\cdots$ **then** ▷ Repeats for other types of terrain
    ⋮ ▷ Assign an empirical cost to it
  end if
  Save t_t under grid cell i.
end for

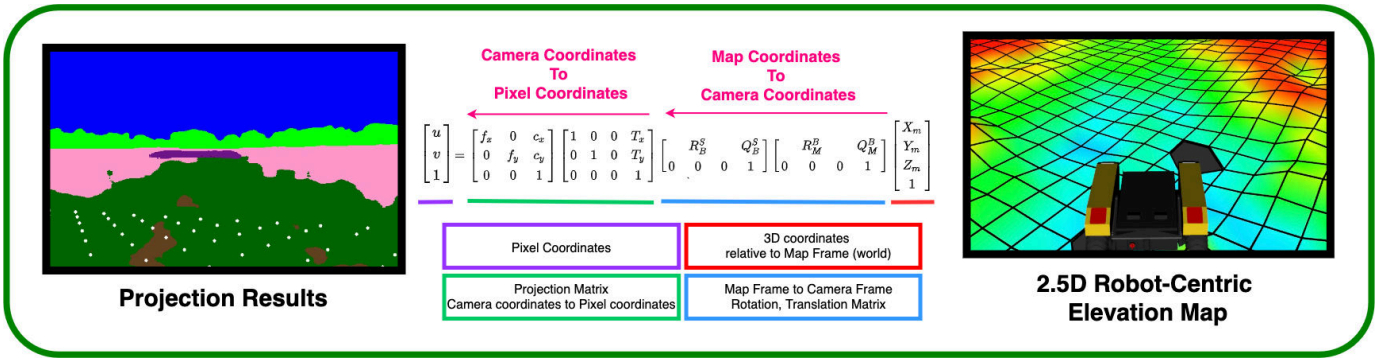

Fig. 2. Illustration of the projection process. Positions of the grid cells are projected on the semantic mask.

To map the terrain locations from the 2D pixel coordinates to the 3D world coordinates and represent it on the grid map, several transformations are performed. First, a request for the latest 2.5D robot-centric elevation map is sent to the elevation mapping module. Once the elevation mapping module has responded to the request and returned the latest elevation map, the 3D position of each grid cell on the map can be accessed and expressed as $P_i = (x_i, y_i, h_i)$, where x_i and y_i are the position of the grid cell i related to the map frame, which is under the world coordinates. And h_i is the estimated height at that position. Since the map frame M is related to the body frame B by r_{BM} and ϕ_{BM}. And the 3D camera coordinate frame is related to the body frame B by a fixed translation r_{BS} and fixed rotation ϕ_{BS}. Therefore, we can transform a 3D point from map frame M to camera coordinate frame S.

For example, the position of cell i related to map frame M can be expressed as:

$$P^M = \begin{bmatrix} X^M \\ Y^M \\ Z^M \end{bmatrix} \tag{1}$$

Let $Q_M^B \in \mathbb{R}^{3x1}$, $R_M^B \in \mathbb{R}^{3x3}$ be the translation vector and rotational matrix between the map frame M and body frame B respectively. And $Q_B^S \in \mathbb{R}^{3x1}$, $R_B^S \in \mathbb{R}^{3x3}$ are the translation vector and rotational matrix between the body frame B and camera coordinate frame S. P^S is the position of the cell related to the camera coordinate frame S.

$$\begin{bmatrix} P^S \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^S & Q_B^S \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} R_M^B & Q_M^B \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} P^M \\ 1 \end{bmatrix} \tag{2}$$

Once the grid positions are represented in the camera coordinate frame S, we can filter out the grids which are behind or out of sight of the camera to avoid redundant computation. Then, Pinhole Camera Model [18] is used to project the 3D grid position in camera coordinate frame S onto the 2D pixel plane of the semantic mask as described by equation 4.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} P^s \\ 1 \end{bmatrix} \tag{3}$$

For monocular camera, $T_x = T_y = 0$. Where c_x and c_y are the principal points and f_x and f_y are the focal lengths. These camera's internal and external intrinsic for a rectified image can only be found by calibrating the camera. (u, v) are the pixel coordinates projected from the 3D position of the grid cell. Fig. 2 illustrates the projection process.

Different colours on the semantic mask represent different types of terrain. Once we know where the grid is projected on the semantic mask, we can retrieve the terrain information from the semantic mask by checking the colour at the projected pixel coordinate.

For each cell i in the filtered elevation map, the 3D cell position $P_{3D} = (x_i, y_i, h_i)$ is mapped to a 2D pixel coordinate $P_{2D} = (u_i, v_i)$. By checking the RGB values at (u_i, v_i), we can know which type of terrain is located at (x_i, y_i). Then we can assign a empirical traversability cost t_t, according to the type of terrains it belongs to, in the range from 0 to 1; i.e. $t_t \in [0, 1]$. $t_t = 0$ indicates fully traversable, and $t_t = 1$ means fully not traversable. In other words, the lower the cost, the more traversable it is. Algorithm 1 delineates the traversability cost assignation according to the semantic segmentation results.

2) Cost map based on terrain geometrical properties: The previous section has emphasized the importance of assessing terrain traversability through its geometrical properties. It accounts for the physical capabilities of the robotic platform in the traversability analysis. The three terrain geometrical properties 1) ***Slope*** s , 2) ***Roughness*** r , and 3) ***Step Height*** h are computed similar to [4], [19] by applying different filters on the elevation map. These properties are all derived from the height information stored in the elevation map. They are combined by equation 4 and output a traversability cost in the range from 0 to 1, i.e. $t_g \in [0, 1]$.

$$t_g = w_1 \frac{s}{s_{crit}} + w_2 \frac{r}{r_{crit}} + w_3 \frac{h}{h_{crit}} \tag{4}$$

While w_1, w_2 and w_3 are the weights sum up to 1. The critical values s_{crit} ,r_{crit} and h_{crit} are robot-specific maximum allowable values. If one of the terrain geometrical properties is greater than its critical value, the traversability cost will be set to 1, meaning fully not traversable. These traversability costs will be also saved under the grids.

TABLE I
PER-CLASS IOU(%) ON THE RELLIS-3D TEST SET OF GATED-SCNN AND ERFNET

Network	Sky	Grass	Tree	Bush	Concrete	Mud	Person	Puddle	Rubble
Gated-SCNN	97.02	84.95	78.52	70.33	83.82	45.52	90.31	71.49	66.03
ERFNet	96.88	89.33	75.14	73.69	79.15	39.30	82.03	64.57	61.69

Barrier	Log	Fence	Vehicle	Object	Pole	Water	Asphalt	Building	**Mean**
55.12	2.92	41.86	46.51	54.64	6.90	0.94	44.18	11.47	**50.13**
30.99	0.26	27.88	14.70	32.26	0.71	7.90	5.19	0.86	**41.19**

3) Fusing both traversability cost maps: In the end, the module will fuse these 2 sets of traversability costs into 1 by:

$$t_f = \alpha \cdot t_t + \beta \cdot t_g \tag{5}$$

Where t_t and t_g are the traversabilty costs based on terrain types and geometrical properties respectively. And t_f is the fused traversability cost adjusted by the weighting factors α and β. It is the final product of the proposed framework.

IV. EXPERIMENTS, RESULTS AND DISCUSSIONS

A. Comparison between Gated-SCNN and ERFNet

Aforesaid, two different deep neural network architectures, Gated-SCNN and ERFNet, were tested for the semantic segmentation task.

1) Evaluation Metrics: The mean intersection-over union (mIoU) metric [20] is widely used for evaluating the neural network performance on semantic segmentation. It is given by:

$$mIoU = \frac{1}{C}\sum_{c=1}^{c}\frac{TP_c}{TP_c + FP_c + FN_c} \tag{6}$$

TP_c, FP_c and FN_c represent the number of true positive, false positive and false negative predictions for each class c. And C is the total number of classes.

2) Per-class accuracy: Table I shows the results on every one of the eighteen classes evaluated on the RELLIS-3D test set at 600×400 for both neural network architectures. Gated-SCNN performed slightly better on the mostly seen classes, such as Sky, Tree, Concrete, Ruddle, and Person. Whilst the dominance is not notable as ERFNet performed better than Gated-SCNN on Grass and Bush. To sum up, for the mostly seen classes, both networks have a very similar performance.

For those classes with rare appearances in the dataset, e.g., Barrier, Log, Fence, Pole, Asphalt and Building, Gated-SCNN performed better than ERFNet significantly. For an instance, Gated-SCNN achieved 44.18% IoU for Asphalt while ERFNet was only able to achieve 5.19% IoU. However, the advantage of Gated-SCNN in these challenging classes are not convincing. Yet, it performed better than the ERFNet. But the achieved IoUs for these classes are still relatively low, many of them are below 50%. Poor performance on these challenging classes is due to the serious class imbalance of the dataset.

3) Real-time performance: Table II shows the forward pass time (fwt) of both network architectures to process a single RGB image in the resolution of 600×400 on a standalone NVIDIA GTX 960. Test set of RELLIS-3D was used for the evaluation. The timer was started once the image had been loaded. Then it was paused once the semantic mask had been generated. Repeating for 1672 times, the mean inference time was calculated. ERFNet only took 0.094 seconds to process an input image and output its semantic mask. While the Gated-SCNN was much slower, it took 0.935 seconds to infer one single image.

TABLE II
FORWARD PASS TIME FOR GATED-SCNN AND ERFNET.

Network	**NVIDIA GTX 960** 600×400	
	fwt(s)	**fps(hz)**
Gated-SCNN	0.935	1.07 ($\approx$ 1)
ERFNet	0.094	10.64 ($\approx$ 10)

In summary, ERFNet is nearly 10 times faster than the Gated-SCNN. Therefore, ERFNet is ideal for using on a mobile robot that aims to understand the surrounding environment as much as possible in a real-time manner.

4) Qualitative Comparison: Fig. 3 shows four examples of semantic segmentation produced by Gated-SCNN and ERFNet. Visually, the performance of both networks is pretty much the same. Both networks can accurately segment the terrain ahead of the robot. For those commonly seen classes such as grass, tree and bush, the semantic masks outputted from both neural networks are nearly identical. The 10% mIoU

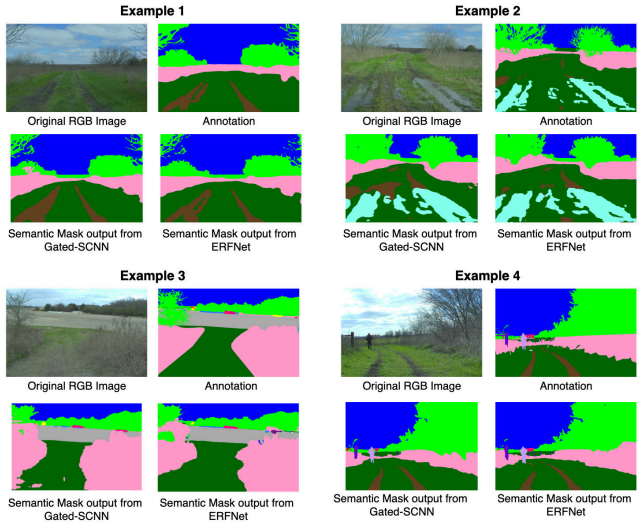

Fig. 3. Qualitative Comparison between Gated-SCNN and ERFNet.

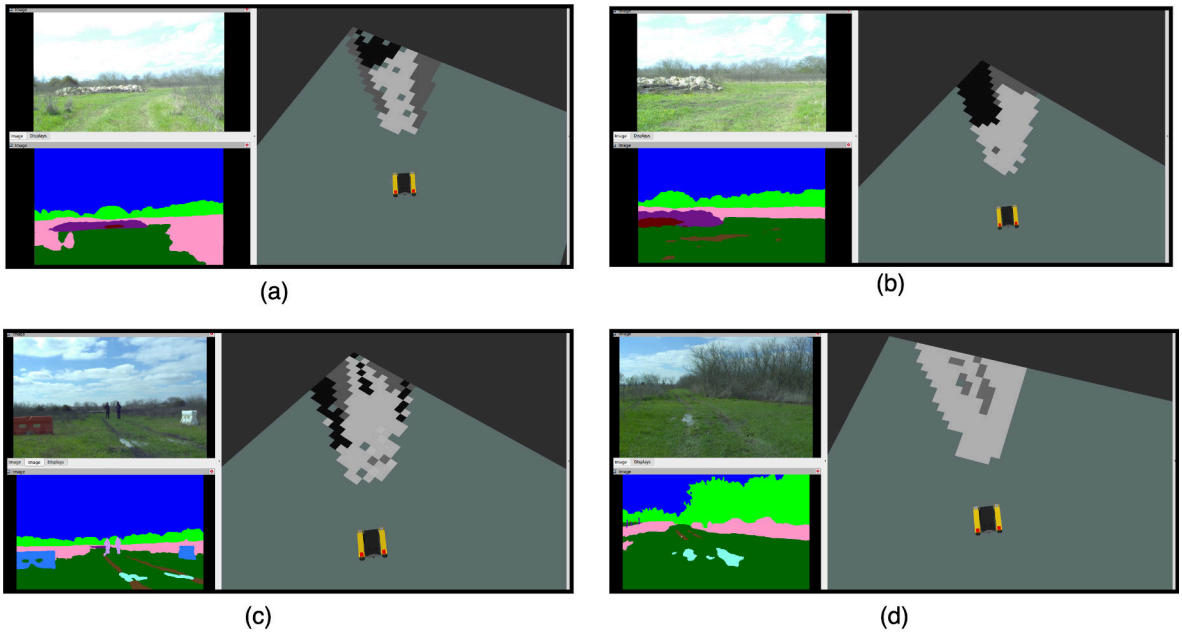

Fig. 4. Visualization of the traversability cost map generated by the proposed framework. In each figure, the image on the top left corner is the RGB image captured by the camera. The image below is the semantic mask output from the semantic segmentation module. The right-hand side of the figure depicts the visualization of the terrain traversability cost map. Grids in deeper colour represent less traversable terrain.

differences between the Gated-SCNN and ERFNet are not notable.

B. Terrain Traversability Map Generation Result

To evaluate and test the proposed framework, the synchronised and real sensor data were used. We captured the data by driving the Warthog platform [22] in different real off-road environments. The Warthog equipped following sensors for gathering the RGB images, LIDAR point cloud and motion information.

1) 1 × RGB Camera: Basler acA1920-50gc camera with 16mm/F18 EDMUND Optics lens, image resolution 1920x1200, 10 Hz
2) 1 × Velodyne Ultra Puck: 32 Channels, 10hz, 40○vertical field of view
3) Inertial Navigation System (GPS/IMU): Vectornav VN300 Dual Antenna GNSS/INS, 300 Hz GPS, 100 Hz IMU

Fig. 4 illustrates the outcomes of the framework. The colour of the grid depends on the traversability cost assigned to that grid. Less traversable areas are assigned with a higher traversability cost. Such that the locations of those less traversable areas are represented by the grids coloured in deeper colour. Oppositely, grids with lighter colours represent the more traversable area.

For example, in Fig. 4(a), Both sides of the robot are covered by bushes. As a result, grids on the left and right are coloured in a deeper colour, visualising the less traversable areas. While there is a large patch of grass in front of the robot and right between the bushes. Traversability of grass is relatively higher, therefore lower traversability costs are assigned to these grids. Thus, these grids are colourised in a lighter colour. While there is a large piece of ruddle behind the grasses. Terrain covered by ruddle is completely not traversable. So that, those grids representing the ruddle are coloured in black.

Another example is shown in Fig. 4(d). The robot sees there are patches of puddles in front. Puddle is less traversable compared to grass. Therefore, the colour of the grids representing the puddles are slightly deeper than the surrounding grids.

V. Conclusion

In this paper, we have presented a framework for terrain traversability analysis that is especially crucial for autonomous navigation in the off-road environment. Semantic segmentation is used to identify different types of terrain ahead of the robot. With the aid of elevation mapping, we acquire the height information of the surrounding terrain. This information allows us to derive the geometrical properties of the terrain: slope, step height and roughness. The traversability costs are assigned and computed based on the terrain types and the geometrical properties. These costs are fused and represented on the 2D robot-centric cost map which is accessible for different path planners, such as Dynamic Window Approach (DWA)[23] or Rapidly-Exploring Random Tree (RRT)[24].

For further improvement of off-road environment semantic segmentation, expanding the dataset with a variety of off-road environments could relieve the imbalance of classes, thus improving the overall performance of the neural network. Optimizing the network performance by altering the network design is also a promising research direction. We also plan to extend our work by incorporating a real-time path plan-

ner into the framework. We believe the well-rounded terrain traversability information could help us to plan an optimal and terrain-aware path.

REFERENCES

[1] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," Eng. Appl. Artif. Intell., vol. 26, no. 4, pp. 1373–1385, 2013.

[2] D. C. Guastella and G. Muscato, "Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review," Sensors (Switzerland), vol. 21, no. 1, pp. 1–22, 2021.

[3] S. Vecherin et al., "Artificial Intelligence and Machine Learning for Autonomous Military Vehicles," no. August, 2020.

[4] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," IEEE Int. Conf. Intell. Robot. Syst., vol. 2016-Novem, pp. 1184–1189, 2016.

[5] Péter Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 3019–3026, 2018.

[6] R. Thakker et al., "Autonomous Off-Road Navigation over Extreme Terrains with Perceptually-Challenging Conditions," Springer Proc. Adv. Robot., vol. 19, pp. 161–173, 2021.

[7] L. Dabbiru et al., "Traversability mapping in off-road environment using semantic segmentation," 2021.

[8] K. Otsu, M. Ono, T. J. Fuchs, I. Baldwin, and T. Kubota, "Autonomous Terrain Classification with Co-and Self-Training Approach," IEEE Robot. Autom. Lett., vol. 1, no. 2, 2016.

[9] B. Rothrock, J. Papon, R. Kennedy, M. Ono, M. Heverly, and C. Cunningham, "SPOC: Deep learning-based terrain classification for Mars rover missions," in AIAA Space and Astronautics Forum and Exposition, SPACE 2016, 2016.

[10] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-Time Semantic Mapping for Autonomous Off-Road Navigation," pp. 335–350, 2018.

[11] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," IEEE Trans. Intell. Transp. Syst., vol. 19, no. 1, pp. 263–272, 2018.

[12] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard, "Deep Multispectral Semantic Scene Understanding of Forested Environments Using Multimodal Fusion," in Springer Proceedings in Advanced Robotics, 2017.

[13] Y. Zhao, P. Liu, W. Xue, R. Miao, Z. Gong, and R. Ying, "Semantic probabilistic traversable map generation for robot path planning," IEEE Int. Conf. Robot. Biomimetics, ROBIO 2019, no. December, pp. 2576–2582, 2019.

[14] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," Proc. IEEE Int. Conf. Comput. Vis., vol. 2019-October, pp. 5228–5237, 2019.

[15] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," Adv. Neural Inf. Process. Syst., vol. 32, no. NeurIPS, 2019.

[16] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "RELLIS-3D dataset: Data, benchmarks and analysis," arXiv, 2020.

[17] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," pp. 248–255, 2009.

[18] J. Ponce and D. Forsyth, Computer vision: a modern approach. 2012.

[19] A. Chilian and H. Hirschmüller, "Stereo camera based navigation of mobile robots on rough terrain," 2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009, pp. 4571–4576, 2009

[20] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," Int. J. Comput. Vis., vol. 111, no. 1, pp. 98–136, 2015.

[21] "Benchmark Suite – Cityscapes Dataset." [Online]. Available: https://www.cityscapes-dataset.com/benchmarks/. [Accessed: 08-Jul-2021].

[22] "Warthog Unmanned Ground Vehicle Robot - Clearpath." [Online]. Available: https://clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/. [Accessed: 05-Jul-2021].

[23] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robot. Autom. Mag., vol. 4, no. 1, 1997.

[24] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," In, 1998.

2022-03-22

Hybrid terrain traversability analysis in off-road environments

Leung, Tiga Ho Yin

IEEE

Leung THY, Ignatyev D, Zolotas A. (2022) Hybrid terrain traversability analysis in off-road environments. In: 2022 8th International Conference on Automation, Robotics and Applications (ICARA), 18 February - 20 March 2022, Prague, Czech Republic

https://doi.org/10.1109/ICARA55094.2022.9738557