Terrain traversability calculation using visual friction estimation and geometric information fusion for sloped terrains

Arijan Vidulin* and Goran Vasiljević*

Abstract— **Traversability analysis remains a significant challenge in achieving fast and reliable robot navigation in unstructured terrains. Often, a single sensor does not provide sufficient information for a satisfactory qualitative assessment of surface conditions, so sensor fusion is proposed as a solution. In this paper, a new method for traversability calculation is presented by utilizing image semantic information for surface friction estimation and localization, alongside existing methods for extracting geometrical features using LiDAR point cloud. Traversability cost is then analytically calculated by combining surface slope value extracted from geometrical features and friction coefficient estimation derived from Large Language Model (LLM) and segmented using Segment Anything Model (SAM3). Traversability is then visualized as a 2D occupancy grid representing a bird's-eye view map of traversability costs for the observed terrain.**

I. INTRODUCTION

In contrast to well-known structured environments like warehouses or urban roads, many factors impact robot locomotion in unstructured terrains. It is difficult to account for every possible situation that could compromise a robot's mission or affect its ability to move and interact with the environment. Additionally, the unpredictability of structure and features in these conditions makes it challenging for robot localization algorithms to work reliably. Many approaches have addressed the challenge of traversability estimation especially in unstructured terrains. In a recent survey article Shu and Dong [1] provided an extensive overview of terrain traversability evaluation, classifying methods based on multiple dimensions including sensor types, robot types, usage scenarios, and learning approaches. Our proposed method focuses on combining geometric information from point cloud, specifically surface slope angle, with material friction estimation from RGB images. By analytically combining these two values, a 2D occupancy map is derived with a range of [0,100], where a higher value represents greater difficulty and lower traversability for a mobile robot in that specific area. The calculated occupancy map can then be used in a local path planner pipeline to generate an optimal trajectory.

The main contribution of this paper is a new analytical method for traversability calculation that combines slope angle extraction from point cloud data [16] with a novel approach to friction coefficient estimation using a combination of a Large Language Model (LLM) and the Segment Anything Model (SAM3). In the proposed analytical method, the friction coefficient and terrain slope angle are integrated through force balance equations for a rigid body on an inclined plane. The goal is to generate a traversability map that encapsulates information about terrain steepness and slipperiness, enabling a robot to identify the optimal route in uneven, material-diverse environments with obstacles of varying slope and surface material. (Fig. 1).

The paper is divided into several chapters: similar research papers on traversability are presented in II, the research methodology is shown in III, and experimental results are provided in IV. A comprehensive conclusion with a focus on future work is given in V.

Fig. 1. Polygon in Isaac Sim shown from multiple perspectives. The mobile robot must analyze different slope angles of polygons and their various surface types to form a comprehensive occupancy map.

II. RELATED WORK

Several survey articles on traversability estimation methods have organized these methods into various categories. Traversability estimation can be defined in different ways depending on the robot's mission, structural design, and the specific environment in which it must operate, and it requires a higher level of abstraction. Sevastopoulos and Konstantopoulos [2] classified traversability estimation techniques into non-trainable, conventional machine learning, and deep learning categories. They further divided the non-trainable methods into stereo vision and grid-based representations,

* Arijan Vidulin and Goran Vasiljević are with University of Zagreb Faculty of Electrical Engineering and Computing, 10000 Zagreb, Croatia
This work was supported by the project Autonomous Robotic Technology for Environmental Monitoring, Intervention and Safety (ARTEMIS) DIGIT.2.1.02.14 financed through the Loan Agreement (Loan No. 9558-HR) between the Republic of Croatia and the International Bank for Reconstruction and Development, which is part of the World Bank Group for the Digital, Innovation, and Green Technology Project (DIGIT Project).

discussing elevation map calculation. In the previously mentioned survey paper, Shu and Dong [1] addressed multi-sensor fusion, and several articles of interest were cited, as this paper focuses on the fusion of LiDAR point cloud and camera image feed.

In our proposed principle, a well-established geometrical method for terrain slope calculation is used. However, to our knowledge, none of the existing methods or articles use semantic information to estimate the friction coefficient from images. In our approach, this information is key to analytically calculating traversability values in the observed environment.

A. Multi-sensor fusion

Zhang and Lyu [3] propose a method that analyzes traversability based on semantic information of terrain using a convolutional neural network (Gated-SCNN), geometric information, and robot mobility. In their paper, each terrain type is represented by the coefficient of semantic information (k) in the traversability analysis, which is later used in the calculation of traversability values.

Saucedo [4] presents an environmentally agnostic method of traversability estimation based on a novel decaying exponential function to fuse semantic information and geometric features extracted from RGB-D images. The baseline algorithm uses two LRASPP MobileNet V3 large CNNs for semantic segmentation of the RGB image: one for terrain type determination and the other for detecting see-through obstacles in the environment. Surface normals are extracted from depth images to calculate terrain inclination. The proposed decay function models the traversability of a surface so that both the slope with each respective semantic label contribute to traversability scores, with each semantic label assigned a different traversability weight, while the function always penalizes the highest slopes.

Sock and Kim [5] present an approach that estimates the traversability of unstructured roads and builds an online 2D probabilistic grid map using 3D-LIDAR and a camera. A linear SVM classifier is used for the detection problem, while LIDAR is used for 2.5D elevation grid map generation, and both types of information are fused under the Bayesian principle.

Zhou [6] proposes a similar method to Sock [5], combining semantic and geometric information from a camera and LiDAR, respectively, where a bottleneck transformer-based deep neural network architecture (GANav) is used to perform traversable area segmentation.

Schilling [7] presents a multi-sensory terrain classification algorithm with a generalized terrain representation using semantic and geometric features, employing a shallow classifier to predict one of three specific traversability classes.

Pan [8] proposes a method that uses only range and inertial sensors to form a dense map with drivable information.

B. Learning-based methods

Shaban [9] presents a deep neural network that directly predicts a local map encoding terrain classes exclusively from sparse LiDAR inputs. Gasperino [10] presents a self-supervised approach for learning to predict traversable paths for wheeled mobile robots. They argue that traction can be estimated for rolling robots using kinodynamic models. Using traction estimates provided by an online receding horizon estimator, they train a traversability prediction neural network in a self-supervised manner. In an improved version [11], by using a sequence of images, the robot's awareness of its surroundings is significantly enhanced, enabling it to predict the traversability of terrains that are not immediately visible. Eder [12] proposed a method in which the cost estimation for different terrain types is based on locomotion data obtained in realistic standardized experiments and earth observation data. In their paper, Waibel [13] proposes two approaches – statistical and learning-based – where IMU data is predicted for traversability estimation. Reutz, in their paper [14], presents a learning-based method that estimates traversability in unstructured environments dominated by vegetation, based on a 3D voxel representation using camera and LiDAR fusion.

As our method differs from the aforementioned research approaches for traversability calculation, there is existing work regarding friction estimation from image data in the automotive context. Jin [15] proposed a deep learning method in which the road surface in the image is segmented and identified to obtain the road type. It is shown that fusing this information with a road friction estimator with tire longitudinal force estimation helps achieve faster convergence and higher accuracy.

III. METHODOLOGY

For analytical calculation of the traversability of the observed terrain, the slope angle value and the friction estimation of the surface are needed. Point cloud geometric data is used to extract the slope value, while the image frame is provided to the LLM to output estimated recognized materials and their respective friction coefficients. The recognized material outputs are then forwarded to SAM3, which precisely segments the given materials and outputs the locations of the segmented areas. Geometric and semantic information are then combined and analytically processed to produce a unified occupancy map of terrain traversability. For better clarity, Table I summarizes the main variables used throughout this section.

A. Slope angle calculation

Geometric feature extraction, specifically slope angle calculation from point cloud data, is performed using the existing traversability method from Saravanan [16]. In their work, Saravanan et al. present an enhanced framework for autonomous 3D exploration that integrates Fisher information and a traversability-aware adaptive roadmap. The method uses the ORB-SLAM3 algorithm for localization, which both local and global traversability calculations use to build a map of the environment. This map is then used to find frontier points in the environment that the mobile robot can reach, based on the calculated traversability. Exploration path

TABLE I

NOTATION OF THE RELEVANT VARIABLES USED IN THE PAPER

Symbol	Description
α_n	Normalized slope angle value
α	Raw slope angle value
$\mathbf{n}_c$	Normal vector of the sloped surface
O_{slope}	Occupancy grid cell value calculated from slope angle value α
O_f	Occupancy grid cell value calculated from friction coefficient value μ
O_{ftot}	Global occupancy grid cell value calculated from merged local friction occupancy maps
c_f	The fused confidence value of combined friction estimations calculation
T	Non-dimensional traversability metric
O_{trav}	Occupancy grid cell value calculated from traversability metric T
O_{cell}	The calculated occupancy grid cell value for the interpolated cell
RMSE	Root Mean Square Error

selection is then initiated to output the optimal path for the local planner to execute.

Environment traversability mapping uses LiDAR point cloud data and calculates statistical features such as slope, step, and roughness, which are computed based on subsets of points within each grid map cell and its neighboring cells. In the referenced research, the final traversability cost value is determined as the maximum among the computed metrics, prioritizing the robot's safety during decision-making.

For our research method, only the slope angle is considered. In [16], the slope is normalized by the maximum slope angle the mobile robot could traverse:

$$\alpha_n = \frac{\arccos\left(\mathbf{n}_c \cdot [0,0,1]\right)}{\alpha_{\max}} \tag{1}$$

where $\mathbf{n}_c$ represents the normal vector of the sloped surface. We modify the equation by extracting only the raw angle value:

$$\alpha = \left|\arccos\left(\mathbf{n}_c \cdot [0,0,1]\right)\right| \tag{2}$$

The absolute value of the function is used to ignore possible edge cases. If, in a later iteration, a larger slope angle is calculated for the same grid cell, the new value is integrated using the weighted average method. Finally, the angle value α is converted to an occupancy grid value with:

$$O_{slope} = 100\frac{\alpha}{1.22} \tag{3}$$

The angle value α is normalized using the maximum estimated traversable angle for the mobile robot, approximately $70°$ $(1.22, \mathrm{rad})$. This value was determined empirically rather than derived analytically. With new slope angle measurements for the existing cell, the slope angle α is averaged, and O_{slope} is then recalculated.

B. Friction coefficient estimation

For surface friction estimation, the RGB image feed from the D455 camera is used. The novel core idea is to use the open-source SAM3 algorithm to segment specific surfaces based on material. However, SAM3 requires word prompts of materials as input, and since the materials present in the frame are *a priori* unknown, an additional step is needed to recognize materials. The image frame is provided as a prompt via API key to the gpt-4.1-mini model, which is requested to output the three most dominant material names it can recognize in the frame, as well as a friction coefficient estimation. The output prompts from the LLM model are sequentially provided to SAM3, which then generates separate image masks for each material with respective confidence scores.

If the LLM generates incorrect material predictions, SAM3 returns an empty mask for the associated prompt because the specified material cannot be segmented in the image. This behavior reduces the risk of false material classifications affecting the occupancy map accuracy.

For each mask, non-transparent pixels are identified, and if a pixel's depth is less than d_{max}, its z component is extracted. Each pixel is then projected to world coordinates using the camera's intrinsic information:

$$x = \frac{(u - c_x)\cdot z}{f_x} \tag{4}$$

$$y = \frac{(v - c_y)\cdot z}{f_y} \tag{5}$$

where x and y are the coordinates of a pixel in the camera frame, u and v are the pixel coordinates in the image, representing width and height, respectively. The variable z denotes the depth value of a pixel, c_x and c_y describe the principal point in the image, and f_x and f_y represent the focal lengths of the camera. Coordinates shown in the camera frame are transformed to world coordinates:

$$[x,y,z]_w^\top = T_c^w \cdot [x,y,z]_c^\top \tag{6}$$

The suffix w denotes coordinates in the world coordinate frame, and the suffix c denotes coordinates in the camera frame, while T_c^w represents the 4x4 transformation matrix that converts coordinates from the camera frame to the world frame.

The occupancy grid value for each pixel is then normalized as follows:

$$O_f = 1 + 99\frac{\mu}{\mu_{max}} \tag{7}$$

The friction value μ is normalized by the highest anticipated friction value μ_{max} to scale values from 1 to 100. Equation (7) ensures that occupancy values are always nonzero for coherent visualization.

C. Global friction map calculation

Local occupancy maps are merged into a consistent global map, where overlapping cells are updated using a weighted average:

$$O_{ftot} = w \cdot O_{ftot} + (1-w)\cdot O_{fnew} \tag{8}$$

$$w = \frac{c_{old}^2}{c_{old}^2 + c_{new}^2} \tag{9}$$

Here, c_{old} represents the combined confidence level for that cell from all previous iterations, and c_{new} represents the confidence level of the most recent estimation. O_{ftot} is the final value of the friction occupancy map cell.

The combined confidence level is updated using the Bayesian rule, taking into account the confidence levels of the most recent SAM3 estimation and the aggregated confidence from previous updates for that cell. The new combined confidence is calculated as:

$$c_f = \frac{P(y|Z_1)P(y|Z_2)}{P(y|Z_1)P(y|Z_2) + (1 - P(y|Z_1))(1 - P(y|Z_2))} \tag{10}$$

where $c_{old} = P(y|Z_1)$, $c_{new} = P(y|Z_2)$, and c_f denotes the fused confidence level after combining the values.

D. Traversability Estimation

To calculate analytical traversability estimation, the algorithm iterates through the slope occupancy grid with every new update to either the friction or slope occupancy map. It transforms slope map indices into world coordinates to locate the corresponding friction data. If no corresponding data exists, the estimation for that point is bypassed, effectively handling the synchronization mismatches between the LiDAR and RGB-D camera. For the existing corresponding data the matching friction value μ is extracted from O_{ftot} in the friction occupancy map for each slope occupancy cell, where α is extracted from O_{slope}. The final traversability value is then calculated as:

$$T = \mu \cdot \cos(\alpha) - \sin(\alpha). \tag{11}$$

This formula is derived from the force equation of a body on a slope assuming static equilibrium while dynamic effects (e.g., robot velocity, acceleration, or wheel slip) are neglected:

$$F_g = mg \cdot \sin(\alpha) \tag{12}$$

represents the force pulling the body down the slope, and:

$$F_c = mg \cdot \cos(\alpha) \cdot \mu \tag{13}$$

is the reactive force acting upward. The resultant force ΔF is normalized to obtain the non-dimensional value T:

$$T = \frac{\Delta F}{mg} = \frac{F_c - F_g}{mg}. \tag{14}$$

The scaled value written in the traversability occupancy map is:

$$O_{trav} = 100(1 - T) \tag{15}$$

where we assume the minimal occupancy value occurs for even terrain and high friction values.

As mentioned in I, a higher occupancy cell value denotes lower traversability and a greater challenge for the robot to traverse that area.

E. Interpolation of cells

The rate of exploration depends on the camera's field of view. To increase the exploration rate without losing traversability accuracy, new cell interpolation algorithm is implemented.

The algorithm iterates through unknown cells in the traversability occupancy map, cells with values of -1, and checks if there is at least one known cell (cell value $\neq -1$) within a 5x5 square around the observed cell (see fig. 2). If so, a weighted interpolation of the cell is performed, using an additional vicinity factor so that cells closer to the observed cell have a greater impact on the calculated traversability value (eq. 17).

$$O_{cell} = \frac{\sum_i C_i \cdot d_i \cdot O_i}{\sum_i C_i \cdot d_i} \tag{16}$$

O_{cell} represents the calculated occupancy value for the interpolated cell, C_i is the confidence level for each surrounding cell that contributes to the interpolation, d_i is its respective distance factor, and O_i is its occupancy value.

$$d_i = \begin{cases} 2, & \text{if } distance = 1 \\ 1, & \text{otherwise} \end{cases} \tag{17}$$

The fused confidence value is then iteratively calculated, similarly to the case of overlapping cells in the global occupancy map update:

$$c_{fi} = \frac{c_{fi} \cdot c_c}{c_{fi} \cdot c_c + (1 - c_{fi}) \cdot (1 - c_c)} \tag{18}$$

where c_{fi} denotes the aggregated confidence value and c_c is the confidence value of the surrounding cell being added to the final result. By implementing this method, map coverage is increased with the same number of observed images, as shown in fig. 3, while maintaining map accuracy as demonstrated in section IV-C.

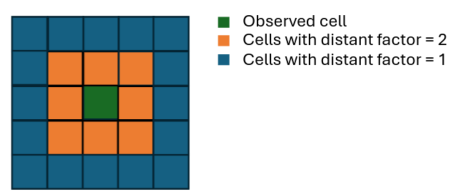

Fig. 2. Cell occupancy estimation regarding it's surrounding cells

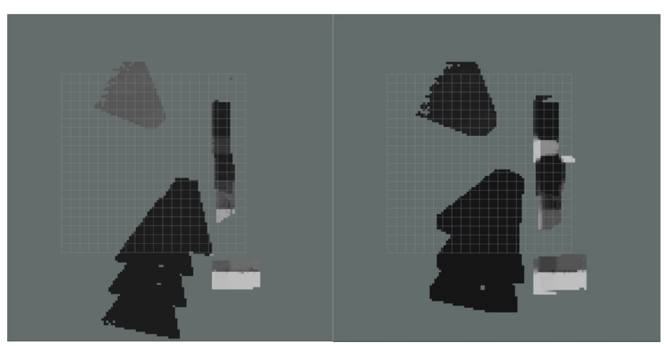

Fig. 3. Comparison of iterations without (left) and with (right) interpolation implementation

F. Maintaining Friction Continuity

To create a more coherent and unified traversability map, new friction continuity method is implemented. If the LLM outputs material estimations containing at least one identical word from the previous iteration, the algorithm constrains the second prompt to use the friction value from the previous iteration. Otherwise, a small change in the prompt results in a slight change in the output friction value (Table II), which visually divides the map but has little impact on the grid values. After unifying friction values for the same material, the final occupancy map is more coherent and robust with respect to surface friction values.

TABLE II
COMPARISON OF FRICTION VALUES FOR SIMILAR PROMPT NAMES

Type	**Friction**
Light brown **wood**	0.4
Beige ply**wood**	0.5
Brown **wood**	0.5
Tan **wood**	0.6

IV. RESULTS

The research and testing in this paper are conducted in the NVIDIA Isaac Sim environment because of its easy integration with ROS2 and its potential for training learning-based models. The NVIDIA Carter mobile robot is used to maneuver through the map while carrying simulated models of the Realsense D455 RGB and depth camera, Ouster VLS128 LiDAR, and IMU unit. The testing arena (Fig. 1) is specifically assembled to test method functionalities, consisting of nine ramp objects grouped in threes. Each of the three blocks of ramp objects has a specific slope value. The first three ramp objects have slope angles of 20.74°, 16.15°, and 8.86°, respectively. Additionally, each ramp object in a block has a unique surface material: white snow, OSB (Oriented Strand Board), and rubber. Each slope-material combination analytically produces a different occupancy value, which is the goal to calculate and present as an occupancy map object.

For the purpose of testing different aspects of the algorithm under repeatable conditions, a ROS bag was recorded while the mobile robot was driven around and across the polygons to capture all relevant areas of the observed map. This approach made it convenient to test both real-time and slowed execution of the algorithm by adjusting the ROS bag playback speed.

In this section, the experimental results of the proposed method are presented. The generation of the ground-truth map is explained, and the validity of the interpolation and continuity methods is evaluated.

When comparing the respective occupancy cell values, the root mean square error (RMSE) metric was used (Eq. 19). Several iterations of the algorithm were performed per experiment to address the variability of LLM and SAM3 outputs, and the average RMSE and its variation were used as metrics of interest.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{O}_i - O_i\right)^2} \tag{19}$$

$\hat{O}_i$ depicts estimated traversability occupancy value for i-th cell, O_i is ground-truth occupancy value for that cell while formula is iterating thorugh n map cells of interest.

A. Ground-truth map generation

To compare experiment outputs with reference values, a synthetic traversability ground-truth map is generated. By combining distance values from Isaac Sim, friction coefficient estimates, and calculated slope angles, a representative occupancy map is created with the same resolution as the experimental maps and in the same reference coordinate frame (Fig. 4).

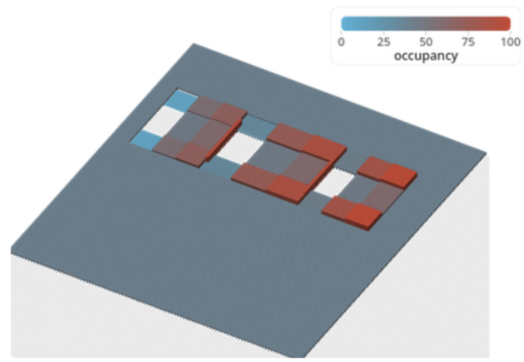

Fig. 4. Colored ground truth occupancy map.

B. Impact of Maintaining Friction Continuity

In the process of friction value estimation and segmentation, only pixels with depth values up to $d_{max} = 10\,\mathrm{m}$ were considered. When calculating the occupancy value based on friction estimation, $\mu_{max} = 1.5$ was used in Eq. (7). Furthermore, world points were binned into a 2D occupancy map with a resolution of $0.25\,\mathrm{m/cell}$, while unknown cells were assigned a value of -1.

Table III presents the impact of maintaining friction continuity on RMSE metrics. The table shows the average RMSE value over 7 iterations and the corresponding standard deviation values. The results indicate that comparable metric values were achieved; however, Figure 5 demonstrates a visual improvement in the final friction occupancy map.

Fig. 5. Comparison of iterations without friction continuity (left) and with active friction continuity (right)

TABLE III

COMPARISON OF RMSE VALUES FOR CONTINUITY

Version	RMSE	σ_{RMSE}
With friction continuity	25.91	2.33
Without friction continuity	24.87	2.44

C. Validity of the interpolation of the unknown cells

To test the cell manipulation method explained in III-E, two types of iterations were performed: one with interpolation enabled and one with interpolation disabled. The categories of interest were RMSE value and cell coverage percentage (Table IV).

TABLE IV

COMPARISON OF RMSE AND COVERAGE PERCENTAGE VALUES

Version	RMSE	σ_{RMSE}	Coverage %	σ_c
Interpolated cells	21.96	3.34	76.69	3.29
Non-interpolated cells	21.27	3.4	63.18	19.34

For each version of the algorithm, 9 trials were conducted on the same ROS bag sequence, and the average metrics of RMSE and coverage percentage were calculated. The results show that the proposed method improved coverage area while maintaining a comparable RMSE value. Possibility of unsafe traversability estimates caused by interpolation in regions where the terrain changes abruptly is reduced to minimum through fusion of the slope calculation with friction coefficient estimation. In these regions, the computed slope angle is high, which consequently yields a low traversability value regardless of possible overestimation of the friction coefficient.

D. Comparison of Real-Time and Slowed Execution of the Algorithm

Although the algorithm was tested on an NVIDIA RTX5070 GPU, a bottleneck occurred while waiting for the LLM response in the pipeline. Solutions tested included slowing the simulation time by a factor of six, which matched the average pipeline delay, or capturing an image every six seconds to maintain real-time performance with a reduced amount of data.

To address the bottleneck, four iterations were completed while running the bag at a rate of 0.17, equivalent to slowing the bag execution by a factor of six. This resulted in the algorithm processing a new image every second, in contrast to real-time execution, where a new image is processed approximately every six seconds.

Both real-time and slowed iterations were performed with cell interpolation enabled and friction continuity active. The recorded sequence included the entire polygon area. The area used for coverage percentage calculation was limited to the slope areas of the map only.

As expected, the coverage results in Table V show that real-time execution covers less area than slowed execution, which processes six times more data. The significant coverage variance, σ_c, in real-time case can also be interpreted as uncertainty regarding whether the algorithm will process an image at the exact moment when the area of interest is recorded. However, the RMSE results in Table V show that slowed execution yields worse final results. This can be attributed to the increased amount of recorded data being compared with the ground-truth labels, which raises the probability of deviations from the ground truth and consequently increases the RMSE.

TABLE V

RMSE AND COVERAGE VALUES IN REAL-TIME VS. SLOWED COMPARISON

Version	RMSE	σ_{RMSE}	Coverage %	σ_c
Real-time execution	26.58	2.15	79.88	24.4
Slowed execution	27.26	2.65	97.38	2.74

Since the future goal is to test the proposed algorithm in a real-world setup, these results are encouraging, as processing image data every six seconds can provide sufficient traversability results (Fig. 6) while enabling local algorithm execution. However, these results are heavily influenced by the mobile robot's motion speed, as the results presented in this article were obtained with a camera moving at 0.5 m/s. This information should be considered when testing the algorithm in a realistic environment.

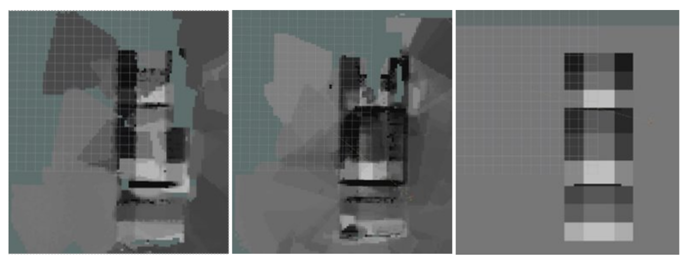

Fig. 6. Comparison of final traversability occupancy maps: real-time iteration (left), slowed iteration (center), and ground truth map (right).

V. CONCLUSION

This paper presents a novel analytical method for traversability calculation by combining slope angle estimation from LiDAR point cloud data with friction coefficient estimation using LLM and SAM3 for semantic segmentation of recognized materials. It is shown that resultant occupancy map achieves strong RMSE values when compared with a synthetic ground-truth traversability map.

In comparison with the existing geometrical traversability method [16], upon which our approach is based, the proposed method provides an additional information layer in which slope angle values are combined with texture recognition data. This enables path planning algorithms to achieve significantly greater robustness when designing robot motion in unstructured terrains. An evident drawback of the proposed method is the substantially higher computational load, which required the algorithm to be executed at a slower runtime or with a lower update rate. Additionally, the LLM introduces an unknown delay ranging from 3 to 6 seconds during the friction coefficient analysis of the image.

Further development of this method includes testing the algorithm on real-world data, such as the Botanic Garden dataset [17], which provides image and point cloud data from unstructured garden terrains. Next steps also involve testing the algorithm on real-world hardware and addressing the bottleneck problem by either running the LLM on a local GPU or optimizing other parts of the algorithm and translating it to C++ to improve execution time. As this algorithm is intended to work alongside path planning methods, it is of interest to create a combined pipeline of both algorithms where path length and duration are optimized together with the selection of the optimal traversable path for the mobile robot to reach the desired goal.

REFERENCES

[1] Y. Shu, L. Dong, J. Liu, C. Liu, W. Wei, Overview of Terrain Traversability Evaluation for Autonomous Robots, *Journal of Field Robotics*, vol. 42, pp. 1724-1765, 2024.

[2] C. Sevastopoulos, S. Konstantopoulos, A Survey of Traversability Estimation for Mobile Robots, *IEEE Access*, vol. 10, pp. 96331-96347, 2022.

[3] W. Zhang, S. Lyu, C. Yao, F. Xue, Z. Zhu, and Z. Jia, Analysis of Robot Traversability over Unstructured Terrain using Information Fusion, *2022 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 413-418, 2022.

[4] M. Saucedo, A. Patel, C. Kanellakis, G. Nikolakopoulos, EAT: Environment Agnostic Traversability for reactive navigation, *Expert Systems with Applications*, Vol. 244, 2024.

[5] J. Sock, J. Kim, J. Min, K. Kwak, Probabilistic traversability map generation using 3D-LIDAR and camera, *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5631-5637, 2016.

[6] L. Zhou, J. Wang, S. Lin, Z. Chen, Terrain Traversability Mapping Based on LiDAR and Camera Fusion, *8th International Conference on Automation, Robotics and Applications (ICARA)*, pp. 217-222, 2022.

[7] F. Schilling, Xi Chen, J. Folkesson, P. Jensfelt, Geometric and visual terrain classification for autonomous mobile navigation, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2678-2684, 2017.

[8] Y. Pan, X. Xu, Y. Wang, X. Ding, R. Xiong, GPU accelerated real-time traversability mapping, *Proceeding of the IEEE International Conference on Robotics and Biomimetics*, pp. 734-740, 2019.

[9] A. Shaban, X. Meng, J.H. Lee, B. Boots, D. Fox, Semantic Terrain Classification for Off-Road Autonomous Driving, *Proceedings of the 5th Conference on Robot Learning*, pp. 619-629, Nov. 2022.

[10] M. Gasparino, A. Sivakumar, Y. Liu1, A. Velasquez, V. Higuti, J. Rogers, H. Tran, G. Chowdhary, WayFAST: Navigation with Predictive Traversability in the Field, 2022.

[11] M. Gasparino, A. Sivakumar, G. Chowdhary, WayFASTER: a Self-Supervised Traversability Prediction for Increased Navigation Awareness, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8486-8492, 2024.

[12] M. Eder, R. Prinz, F. Schöggl, G. Steinbauer-Wagner, Traversability analysis for off-road environments using locomotion experiments and earth observation data, *Robotics and Autonomous Systems*, Vol. 168, 2023.

[13] G. G. Waibel, T. Löw, M. Nass, D. Howard, T. Bandyopadhyay, P. Borges, How Rough Is the Path? Terrain Traversability Estimation for Local and Global Path Planning, *IEEE Transactions on Intelligent Transportation Systems*, How Rough Is the Path? Terrain Traversability Estimation for Local and Global Path Planning, Vol.23, No.9, pp. 16462-16473, 2022.

[14] F. Ruetz, P. Borges, N. Suenderhauf, E. Hernandez, T. Peynot, Forest Traversability Mapping (FTM): Traversability estimation using 3D voxel-based Normal Distributed Transform to enable forest navigation, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8714-8721, 2022.

[15] D. Jin, B. Leng, X. Yang, L. Xiong, Z. Yu, Road Friction Estimation Method Based on Fusion of Machine Vision and Vehicle Dynamics, *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1771-1776, 2020.

[16] S. Saravanan , A. Bains, C. Chanel , D. Vivet, FIT-SLAM 2: Efficient 3D exploration with Fisher information and traversability-based adaptive roadmap, *Robotics and Autonomous Systems*, Vol.194, 2025.

[17] Y. Liu, Y. Fu, M. Qin, Y. Xu, B. Xu, F. Chen, B. Goossens, P. Sun, H. Yu, C. Liu, L. Chen, W. Tao, H. Zhao, BotanicGarden: A High-Quality Dataset for Robot Navigation in Unstructured Natural Environments, *IEEE Robotics and Automation Letters*, Vol.9, No.3, pp. 2798-2805, 2024.