2022 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
AUTONOMY, ARTIFICIAL INTELLIGENCE & ROBOTICS TECHNICAL SESSION
AUGUST 16-18, 2022 - NOVI, MICHIGAN

CREATION OF A GROUND SLOPE MAPPING METHODOLOGY WITHIN THE ROBOTIC TECHNOLOGY KERNEL FOR IMPROVED NAVIGATION PERFORMANCE

Jackson Ramsey[1], Robert Brothers[1], Joseph Hernandez[1]

[1]Southwest Research Institute®, San Antonio, TX

ABSTRACT

This paper presents a new terrain traversability mapping method integrated into the Robotic Technology Kernel (RTK) that produces ground slope traversability cost information from LiDAR height maps. These ground slope maps are robust to a variety of off-road scenarios including areas of sparse or dense vegetation. A few simple and computationally efficient heuristics are applied to the ground slope maps to produce cost data that can be directly consumed by existing path planners in RTK, improving the navigation performance in the presence of steep terrain.

Citation: J. Ramsey, R. Brothers, J. Hernandez, “Creation of a Ground Slope Mapping Methodology Within the Robotic Technology Kernel for Improved Navigation Performance,” In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 16-18, 2022.

1. INTRODUCTION

Military unmanned ground vehicles (UGVs) face a wide variety of off-road terrain features that are difficult to perceive and navigate safely. Conventional techniques for perceiving terrain in commercial autonomy systems assume a locally flat and smooth ground plane, simplifying the problem of segmenting the traversable and non-traversable space. This flat-world assumption works even in cases of steep grade if the UGVs control systems compensate for grade as a normal disturbance input; however, the assumption breaks down in off-road environments where it may be desirable to navigate cautiously on areas of steep grade or otherwise incorporate this information into a path planner that can intelligently choose between navigating steep and flat terrain.

The Ground Vehicle Systems Center (GVSC) Robotic Technology Kernel (RTK), a Robotic Operating System (ROS)-based library of modular software packages used for autonomous navigation in off-road environments, provides multiple path planners that can easily incorporate terrain gradient information to improve their existing off-road navigation capability for UGVs. Navigation performance is heavily dependent on the quality of perception information provided by the autonomy system. RTK currently relies on Light Detection and Ranging (LiDAR) sensors for many perception tasks, including the detection of traversable and non-traversable (i.e., ground and non-ground) areas and segmenting them into maps. Some naïve approaches to generating terrain gradient maps, or slope maps, from the segmented ground map showed promise in simulation and on flat, smooth surfaces such as an asphalt road; however, the presence of either sparse or dense vegetation, which is common for many off-road scenarios, gave very inaccurate and unpredictable results. To solve this problem within the RTK architecture, SwRI developed a novel method for slope mapping that rejects the kind of noise induced by vegetation and thereby enables improved off-road navigation performance.

2. BACKGROUND

The autonomy literature contains many approaches to off-road traversability analysis for UGVs, from simple heuristics to end-to-end machine learning. Each approach makes tradeoffs in precision, computational efficiency, and modularity. The approach taken in this paper most closely resembles the simpler heuristic-based methods to maximize the computational efficiency and modularity properties which are desirable for current RTK platforms.

Machine learning techniques are increasingly popular tools for solving problems in automated driving. Even some older, classic machine learning techniques like support vector machines (SVMs) have been successfully applied to traversability analysis. McDaniel et al. [1] used an SVM classifier to roughly segment LiDAR point clouds into "ground" points and discard extra points associated with low vegetation. Nguyen et al. [2] show a much more complex end-to-end approach using modern deep learning frameworks to build a multi-modal model that directly learns steering output commands from LiDAR point cloud and RGB camera inputs. Guastella and Muscato [3] provide many more examples of machine learning techniques applied to navigation in off-road or other unstructured environments in their extensive survey. All these methods provide promising results, but they require extensive training and must be tailored to the specific UGV they operate on. Additionally, many state-of-the-art machine learning techniques require computing hardware that is not available for all RTK platforms to run efficiently for navigation.

Other researchers have developed many computationally efficient heuristics for filtering unwanted environmental components from LiDAR data. Andujar et al. [4] noted that differences in LiDAR intensity returns could help differentiate between different types of vegetation and soil. This result implies that for specific operating environments it may be possible to discriminate between vegetation and ground simply using intensity thresholds, but this solution would not generalize well for a versatile autonomy system like RTK. Goodin et al. [5] provide a method for calculating traversability based on soil condition, vegetation density, surface roughness, and surface slope. The surface slope and the surface roughness are calculated using LiDAR data; however, soil condition and vegetation density must be known for a given environment or estimated from another source, making the metric inappropriate to use directly in RTK. Shan et al. [6] propose a traversability mapping approach using a Bayesian generalized kernel to fill in sparse surface data from a height map and calculate a traversability metric using step heights, surface slope, and surface roughness. Their approach is similar in nature to the traversability mapping method proposed in this paper.

In this work, we propose a traversability mapping method with a heuristic for vegetation detection. This method uses LiDAR-generated height maps to generate slope measurements around the vehicle. These slope measurements and measurement uncertainty values are combined in a map update step to produce a filtered slope map. The filtered gradient of the surface roughness is then used to determine areas that are likely real slopes in the terrain versus areas that appear to have large changes in slope due to vegetation. This slope map data is then fused into an existing cost map used by path planners in RTK to improve navigation capabilities.

3. SLOPE MODELING

Existing vehicle autonomy hardware architectures present several unique constraints that preclude many common approaches. Unless a dedicated machine exists, the computer running the slope computations likely runs many other processes. Minimizing computational effort of new processes running on this machine must be prioritized to keep other critical processes running as expected. Many of the machine learning approaches are computationally intensive and would fail this constraint. Additionally, these computations must be robust to expected errors and inaccuracies in the system.

Errors can be introduced at several points in typical LiDAR processing and localization algorithms for autonomous vehicles. LiDAR segmentation algorithms filter out certain noise and obstacle points, but occasionally some low object points can slip through the filter. This environment segmentation error is most often observed with dense or cluttered vegetation, where differentiating between thick undergrowth and ground is challenging. Additionally, the vehicle may experience short-term pitch and roll movements that are not reflected in the localization. Subsequent point clouds may then be misaligned. Small pitch and roll errors may also accumulate over time, so that if the vehicle navigates in a loop, overlapping point clouds may be offset vertically.

3.1 Proposed Method

The method used to generate slopes ensures protection from pitch and roll errors by considering each point cloud individually and computing instantaneous slopes. The LiDAR processing pipeline sorts LiDAR points into ground and object point clouds, and only the ground point cloud is used for slope computation.

When a ground point cloud is received, the data is first discretized into a 2D grid of cells centered on the vehicle, where the ground height of a cell, *j*, is the average of the ground height of all points within the cell. Note that, in standard operation, each cell has a side length of 0.3m.

$$\boldsymbol{h_i} = \frac{\sum \boldsymbol{h_j}}{\boldsymbol{n_j}} \tag{1}$$

The point cloud usually contains data in several sparse rings, as the LiDARs send pulses in discrete vertically-separated rings. The heights of empty cells between the rings must be approximated. Various hole filling methods exist; however, approaches that perform a weighted average of all filled cells within a local neighborhood may be prohibitively expensive, as the local neighborhood must have filled cells around the empty cell for an accurate average. Instead of examining the entire neighborhood, we examine the cells in a direct line left and right of the empty cell, and above and below the empty cell in the height map grid. If full cells can be found within a user-configurable distance both above and below, or left and right, the weighted average of the filled cells is used for the height of the empty cell. If both left/right and above/below pairs are filled, the weighted average of the interpolated heights is used.

The operation is performed twice to ensure adequate hole filling. While this hole filling method is computationally quick, it can propagate errors due to single inaccurate cells. To compensate, a uniform smoothing operation is performed on the instantaneous height map with a 3x3 kernel. The instantaneous height maps are not aggregated directly, as roll and pitch drift can lead to discontinuous edges between heightmaps. Instead, the gradient of the instantaneous heightmap is computed using a 3x3 Sobel operator.

$$\boldsymbol{\partial_x} = \begin{bmatrix} \mathbf{-1} & \mathbf{0} & \mathbf{1} \\ \mathbf{-2} & \mathbf{0} & \mathbf{2} \\ \mathbf{-1} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{2}$$

$$\boldsymbol{\partial_y} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{-1} & \mathbf{-2} & \mathbf{-1} \end{bmatrix} \tag{3}$$

The 3x3 Sobel operator, while primarily used for edge detection in images, here produces accurate computations of the ground slope.

The instantaneous gradients are then aggregated with a one-dimensional Kalman-style update. The confidence of the instantaneous slope is estimated with an experimentally determined heuristic function of the distance d from the vehicle to the cell under consideration. Note that the final term models decreased accuracy further than 20 meters from the vehicle.

$$\boldsymbol{v_{new}(d) = 1 + 0.1 * d + 5 * \max(0, d - 20)} \tag{4}$$

The aggregated slope is computed by first finding the Kalman gains k_m and k_a for the magnitude and angular components of the slope from the corresponding variances v_m and v_a at time i.

$$\boldsymbol{k_{m,i} = \frac{v_{m,i}}{v_{m,i} + v_{new}}} \tag{5}$$

$$\boldsymbol{k_{a,i} = \frac{v_{a,i}}{v_{a,i} + v_{new}}} \tag{6}$$

The aggregated slope magnitude m_i and angle a_i are then obtained with the standard Kalman update step from the previous time step and new measurements m_{new} and a_{new}.

$$\boldsymbol{m_i = m_{i-1} * (1 - k_{m,i}) + m_{new} * k_{m,i}} \tag{7}$$

$$\boldsymbol{a_i = a_{i-1} * (1 - k_{a,i}) + a_{new} * k_{a,i}} \tag{8}$$

The variance for the next timestamp is then computed.

$$\boldsymbol{v_{m,i+1} = (1 - k_{m,i}) * v_{m,i}} \tag{9}$$

$$\boldsymbol{v_{a,i+1} = (1 - k_{a,i}) * v_{a,i}} \tag{10}$$

Aggregating gradients instead of direct heights confers robustness against abrupt inaccuracies in roll and pitch as well as long-term drift. The slopes computed in this manner require several observations to be detected and to protect against exaggerated slopes due to object points being erroneously included in the ground point cloud. In practice this method has generated accurate slope values in a wide variety of environments. The slope data is output in a layered cost map for visualization and debugging, with slope magnitude and direction represented in different layers.

3.2 Lethal Slope Cost Map

Accurate slopes can be used for a wide variety of path following and navigation uses. The most basic navigation use-case is to identify regions where terrain is too steep to traverse. To serve this use-case, a cost map is published with obstacles corresponding to cells with a slope magnitude above a user-configurable threshold.

Opening and closing image morphological operations are applied to the lethal slope cost map using a kernel size that is approximately the size of the vehicle to clean up any outlier obstacles and ensure that feature sizes are appropriate for planning. This lethal slope cost map effectively identifies slopes too steep to traverse. Terrain structures smaller than the kernel size are designed to be segmented as objects and handled accordingly; however, traversable areas with dense vegetation can appear to be steep slopes, and thus exclude valid areas. While improving the LiDAR hardware and accuracy of the segmentation algorithm in differentiating between vegetation and ground would help, if the vegetation is sufficiently dense, no LiDAR points will penetrate to the ground. We thus implement a heuristic to identify regions where vegetation of increasing height generates false slopes.

The LiDAR point clouds contain surface roughness data, a measure of the smoothness of the ground in the range [0,1]. It is discretized into cells, hole filled, and smoothed identically to the ground height. However, roughness is aggregated directly via 1-dimensional Kalman update, and the roughness gradient is taken using 3x3 Sobel operator on the aggregated roughness. The roughness corresponds approximately to vegetation depth, and a high roughness gradient to areas where vegetation depth is increasing. For a lethal slope, if the roughness gradient magnitude is above a tunable threshold, and the roughness gradient aligns sufficiently with the slope gradient, then the slope is likely false and is filtered out of the lethal slope cost map.

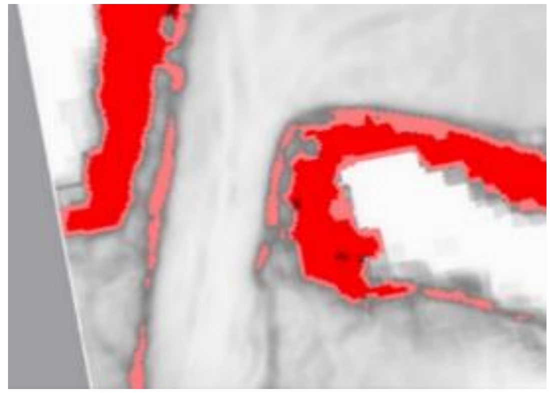

Figure 1: Vegetation-caused false slopes (pink) filtered from lethal slopes (red)

In practice, this heuristic proves highly effective at filtering our false slopes. It is however prone to two types of errors. It may erroneously filter out actual slopes if they contain vegetation of increasing height growing on top, or it may fail to completely filter out false slopes. Both cases have been observed while testing in unstructured environments with complex vegetation, but their occurrence is rare enough that this filtering is considered an improvement on the unaltered lethal slope map.

The lethal slope cost map is sent to a cost map aggregation process to be fused with other cost maps normally produced in RTK. This aggregated cost map can then be consumed by the navigation system and utilized in existing path planners.

4. EXPERIMENTAL TESTING

Slope generation was tested in a suburban street and several unstructured offroad environments. A set of tests are presented here to demonstrate the behavior in varied environments. In each test, a camera image of the environment conditions is present along with a component of the slope map generated in that environment. The slope map images represent the magnitude of the terrain gradient calculated in a 50 meter by 50 meter grid around the vehicle. Traversable slopes are encoded with a grayscale value, with slope magnitude increasing from light to dark gray. All slopes above the user-configurable lethality threshold, set here to 16.7°, are highlighted in pink or red. Potentially traversable vegetation is highlighted by the light pink regions in the map and untraversable slopes are highlighted in red.

Test 1: Suburban Street

In the first test scenario, the vehicle was driven on a two-lane suburban road. An image of the road to the left side of the vehicle and the corresponding slope magnitude map are shown in Figure 2 and Figure 3 respectively.

Figure 2: Suburban street test environment

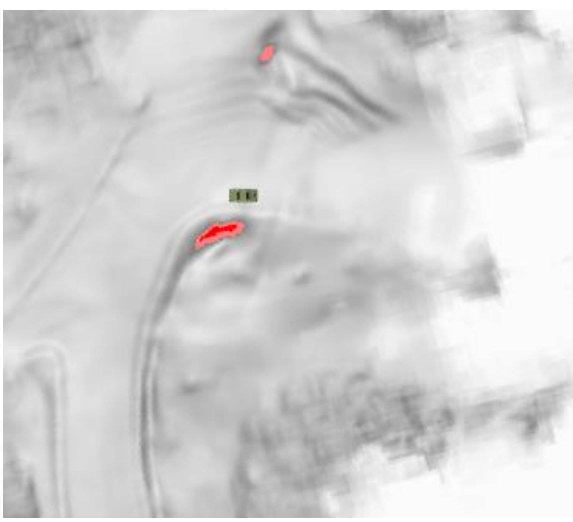

Figure 3: Slope magnitude map for the suburban street environment

In the slope magnitude map, numerous environment features can be observed. Curbs appear as traversable grade along the edges of the smooth road. The map correctly identifies a drainage ditch downhill from the vehicle as an untraversable, or lethal, area.

Test 2: Large Slopes, Minimal Vegetation

The second test scenario shown in Figure 4 and Figure 5 is an offroad environment with sparse vegetation and increasing grade on either side of the vehicle.

Figure 4: Test environment with large slopes and minimal vegetation

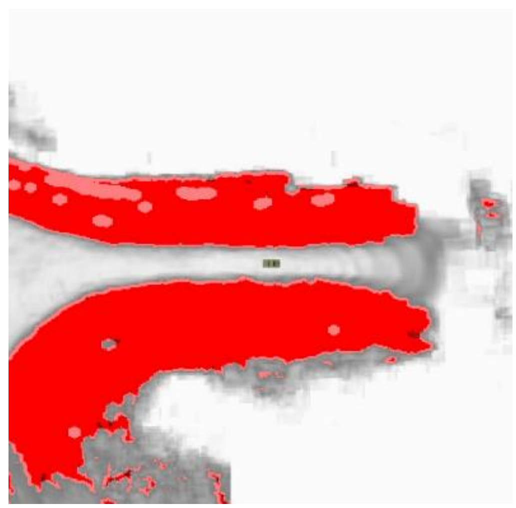

Figure 5: Slope magnitude map for test environment with large slopes and minimal vegetation

The large slopes in the top and bottom of the slope map in Figure 5 are clearly visible and considered untraversable by the vehicle. In this case, the drivable corridor is highlighted well by the slope map directly. Augmenting an off-road path planner with this information would likely keep the vehicle centered on the same path a human driver would naturally take.

Test 3: Large Slopes, Substantial Vegetation

The third test scenario, shown in Figure 6 and Figure 7 depicts an area with large impassible berms, covered in dense vegetation, that surround a small square clearing.

Figure 6: Test environment with large slopes and substantial vegetation

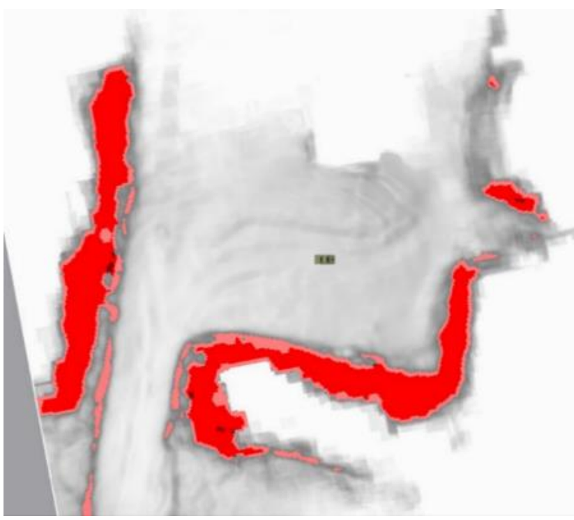

Figure 7: Slope magnitude map for test environment with large slopes and substantial vegetation

The top area of the slope map in Figure 7 appears empty because of obstacle occlusions in the LiDAR data, an intrinsic limitation of the sensor. Note that the slope map correctly identifies the grassy non-slope areas in the paths near the entry and exit to this area. These cells are identified correctly as drivable terrain using the proposed filtering method but would be misclassified as impassible slopes or "bumps" when using only the height map gradient information. The berms are clearly identified and bounded in this environment for areas with valid LiDAR returns.

Test 4: Cluttered Slopes

The fourth test scenario was conducted in an unstructured off-road environment with cluttered vegetation of varying heights and traversability. This environment is shown in Figure 8 and Figure 9.

Figure 8: Test environment with cluttered terrain

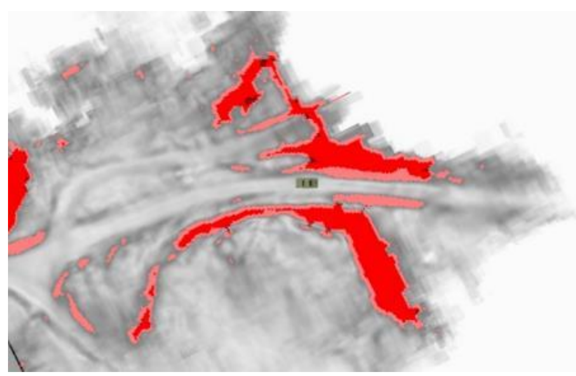

Figure 9: Slope magnitude map for test environment with cluttered terrain

Note that some false slopes appear on the boundary between the vegetation and actual terrain grade in this environment. This test demonstrates a limitation of the vegetation filtering, though erroneous regions usually occur in heavy vegetation and are avoided as normal obstacles or high-cost areas in the cost map.

Test 5: Negative Slopes

In the fifth test scenario, the vehicle was driven on a large natural bridge area with steep, negative grade on either side.

Figure 10: Test environment with negative slopes on either side of the vehicle

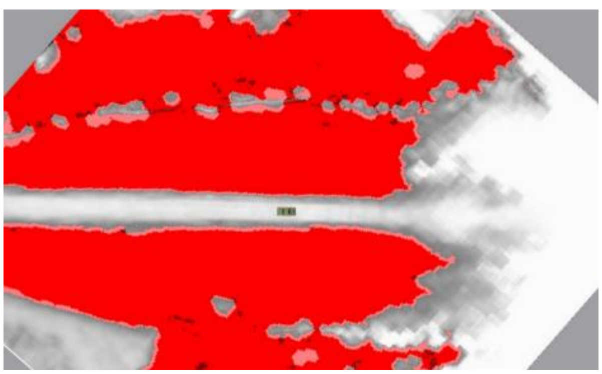

Figure 11: Slope magnitude map for test environment with negative slopes

This test clearly demonstrates that negative slopes are detected and could be planned around, though their visibility is limited due to the LiDAR positions on the vehicle.

Test 6: Path Planning Around Lethal Slopes

In the sixth test scenario, the vehicle was given a waypoint mission in an area with both traversable and untraversable terrain. The goal waypoint was intentionally set on the opposite side of a large berm, visible in Figure 12, from the RTK vehicle's starting position. The Maverick path planner was used to plan to the goal waypoint, utilizing the traversability information provided by the slope map as a part of the planner's normal input. Maverick is an RRT*-based anytime planner that respects vehicle kinematic constraints and is capable of fast replanning [7]. This planner has been extensively tested in RTK and works in a variety of off-road scenarios.

Figure 12: Maverick slope planning test environment

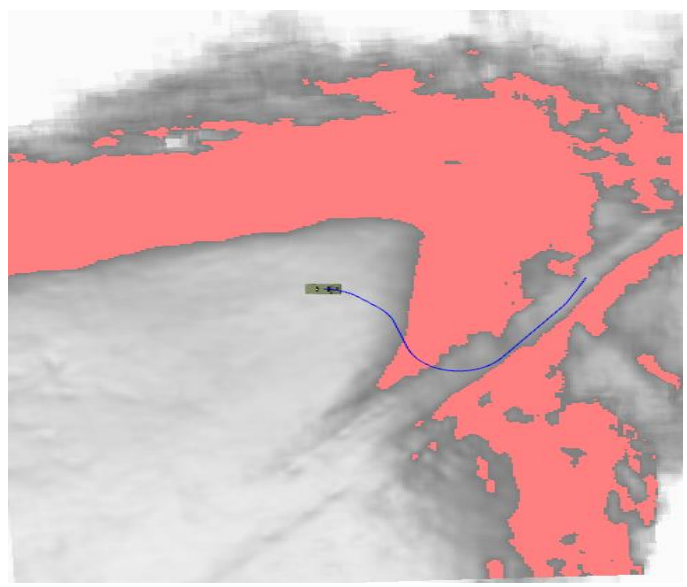

Figure 13: RTK cost map and overlayed slope map with Maverick plan through lethal slope

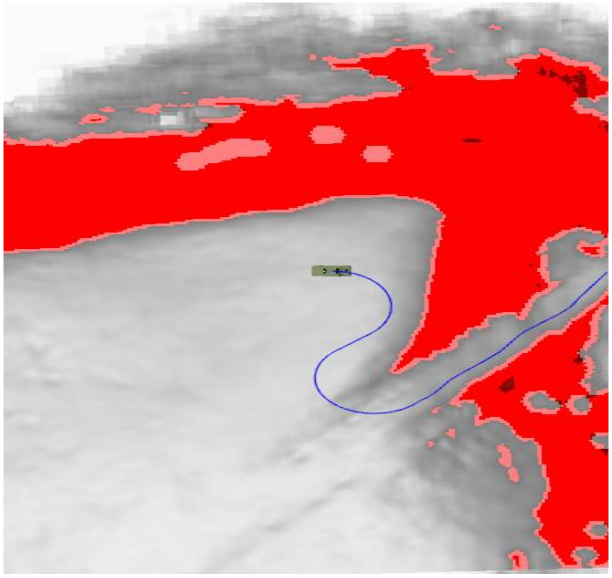

Figure 14: Slope magnitude map and Maverick plan to avoid lethal slope

The maps shown in Figure 13 and Figure 14 contain the Maverick planner's generated path from the vehicle start position to the goal waypoint position as the curve in blue. Figure 13 shows the slope map for the test environment with all slopes artificially considered non-lethal. Without incorporating the slope data as a cost, Maverick plans directly through an area usually classified as lethal slope. Figure 14 shows the same test scenario, but with the slope information included in the cost map sent to Maverick. The generated plan for this run now avoids the lethal slope area completely. This test demonstrates that the slope map information successfully improves navigation performance for an existing path planner in RTK.

5. CONCLUSIONS

Existing Progress

The method outlined in this paper accurately models the terrain slopes in environments with varied amounts of vegetation and complexity. Small features such as curbs are visible, larger hills are represented, and sufficiently steep slopes are treated as obstacles and avoided. Erroneous slopes in the map appear due to dense vegetation but are filtered out of the lethal slope cost map.

Future Work

The traversability mapping method has room for improvement. The computational load could be further reduced by utilizing optimized image convolution routines. The vegetation filtering heuristic can be improved, although more advanced LiDAR and changes to the ground segmentation pipeline may render such improvement unnecessary.

The slope data can also be integrated into many other parts of the RTK system for improved behavior. Slopes can be used to inform path planning, as certain slopes must be traversed at slower speeds. The motion execution modules could factor slopes into acceleration commands, to better summit slopes and minimize orthogonal acceleration during turns. New advanced behaviors could use slopes to identify regions to use as cover, to block line-of-sight from adversaries, or to better survey unexplored regions.

6. REFERENCES

[1] M. W. McDaniel, T. Nishihata, C. A. Brooks and K. Iagnemma, "Ground Plane Identification Using LIDAR in Forested Environments," in *IEEE International Conference on Robotics and Automation*, Anchorage, 2010.

[2] A. Nguyen, N. Nguyen, K. Tran, E. Tjiputra and Q. D. Tran, "Autonomous Navigation in Complex Environments with Deep Multimodal Fusion Network," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, 2020.

[3] D. C. Guastella and G. Muscato, "Learning-Based Methods of Perception and Navigation for Ground Vehicles in Unstructured

Environments: A Review," *Sensors,* vol. 21, no. 1, p. 73, 2021.

[4] D. Andujar, V. Rueda-Ayala, H. Moreno, J. R. Rosell-Polo, A. Escola, C. Valero, R. Gerhards, C. Fernandez-Quintanilla, J. Dorado and H.-W. Griepentrog, "Discriminating Crop, Weeds and Soil Surface with a Terrestrial LIDAR Sensor," *Sensors,* vol. 13, pp. 14662-14675, 2013.

[5] C. Goodin, L. Dabbiru, C. Hudson, G. Mason, D. Carruth and M. Doude, "Fast Terrain Traversability Estimation with Terrestrial Lidar in Off-road Autonomous Navigation," in *SPIE 11758, Unmanned Systems Technology XXIII*, 2021.

[6] T. Shan, W. Jinkun, B. Englot and K. Doherty, "Bayesian Generalized Kernel Inference for Terrain Traversability Mapping," in *Conference on Robotic Learning*, Zürich, 2018.

[7] N. Seegmiller, J. Gassaway, E. Johnson and J. Towler, "The Maverick Planner: An Efficient Hierarchical Planner for Autonomous Vehicles in Unstructured Environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, 2017.