Risk-aware Model Predictive Control for Unmanned Ground Vehicles

Frank Gonzalez, Varun Murali, Fernando Cladera,
Camillo J. Taylor, Vijay Kumar

GRASP Lab, University of Pennsylvania,
{frankgon,mvarun,fclad,cjtaylor,kumar}@seas.upenn.edu

Abstract. Unmanned Ground Vehicles (UGVs) have been successfully deployed in both research and industrial applications in structured settings but are increasingly deployed in unstructured off-road settings. In this context, motion planning in the presence of risks is crucial. Traditionally, mobile robotics has focused on positive obstacles, stationary or moving, as the main source of risks. In this work, we focus on explicitly quantifying and avoiding negative obstacles in addition to positive ones. Negative obstacle detection in UGVs proves to be a key problem for autonomous navigation, as negative obstacles do not register on typical navigation sensors. Traditional obstacle detection techniques fail to identify these obstacles as dangerous, and therefore fail to incorporate them when generating obstacle maps. In this work, we propose a risk-aware planner that is able to combine different sources of risk. We focus on the problem of negative obstacle detection and outline steps for incorporating other risks into our planner. Our results show that negative obstacles are detected with mean Intersection over Union, Precision, Recall, and F1 scores of 0.54, 0.64, 0.77, and 0.69, enabling risk-avoidant navigation around ditches, curbs, and steep drop-offs. The system demonstrates adaptive path planning that circumvents high-risk areas, though performance is limited by occlusion effects and terrain sensitivity. The code of our planner and experimental data is available open source at `github.com/KumarRobotics/risk_mpc`.

Keywords: Field Robotics; Sensing and Navigation; Planning.

1 Introduction

UGVs are central to modern robotics research and industry, enabling autonomous navigation in unstructured, uncertain environments. From agriculture and defense to search and rescue, UGVs must traverse complex terrain where safety, reliability, and adaptability are critical. In this work, we build a framework for risk-aware motion planning. Past work has approached traversability estimation through model-based methods [9, 4] and learning-based techniques that infer region traversability using predefined or implicit semantic affordances [8, 6] but detecting negative obstacles and other unobservable risks remains a significant

challenge. We consider motion planning with both observed obstacles (encoded as constraints) and inferred obstacles (encoded as risk) in a layered framework.

Safe navigation requires accurate terrain assessment using onboard sensors like LiDAR and stereo vision. Early methods relied on Digital Elevation Maps (DEMs), using features such as slope and roughness to classify terrain. Geometry based approaches dominate traversability estimation, using plane fitting, slope, roughness, and uncertainty analysis. Negative obstacle detection, *i.e.* obstacles that protrude or recess from the ground, remains challenging due to their indirect observability; hybrid approaches combine geometric heuristics with machine learning to improve performance. Recent methods [10, 2, 7, 5] leverage learned representations to predict traversability in complex terrain but often focus on positive obstacles and require extensive training data. [1] highlights difficulties in perception when geometric cues are absent, such as ditches concealed by foliage. Appearance-based methods use texture and color cues for terrain classification [1], offering additional information where geometry is insufficient, though they remain sensitive to lighting and occlusions. Challenges persist in modeling uncertainty and achieving real-time performance, making traversability estimation a cornerstone of risk-aware planning. Our contributions are a unified risk-aware planning framework that combines different sources of risk by integrating both positive and negative obstacles into a single planning system, and a model-based negative obstacle detection method that uses geometric analysis of LiDAR scan patterns to identify deviations from expected ground plane geometry. We present a layered Model Predictive Control (MPC) architecture that generates dynamically feasible trajectories while incorporating risk through modified $A^\star$ cost functions, and provide experimental validation on a Clearpath Jackal platform in both semi-urban and rural environments. The implementation and data are available open-source to enable further research in risk-aware autonomous navigation.

2 Technical Approach

Our system consists of various components such as a odometry, perception, planning, and control package (shown in Fig. 1. Communications and data handling is facilitated by The Robotic Operating System (ROS) 2. For odometry, we use Direct LiDAR Inertial Odometry (DLIO) [3] which takes in the raw point cloud from the Ouster and outputs odometry estimates at 100 Hz. The point cloud and odometry is taken in by the perception package, which is responsible for generating risk maps of the environment at 10 Hz. These risk maps are then used by the planning package to generate a coarse path and high-fidelity motion plan, which the control package uses as a reference to generate control actions. The reference trajectory generation nominally operates at 1 Hz. The controller executes at 15 Hz. The reference trajectory and the control action are computed by solving optimization problems formulated with CasADi[1].

[1] https://web.casadi.org

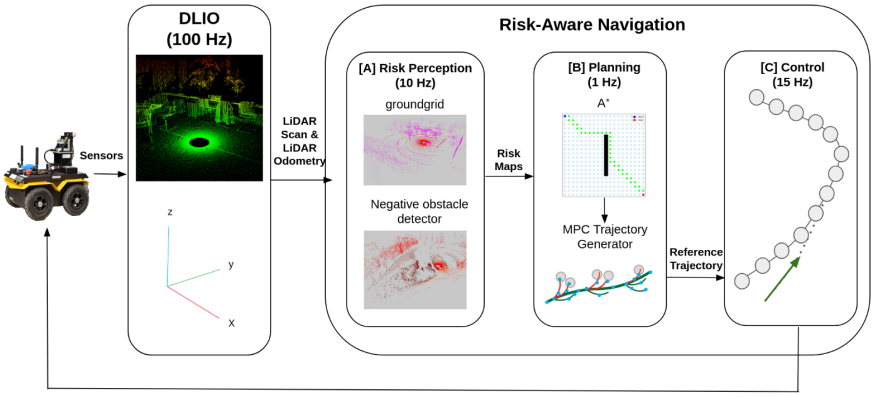

Fig. 1: Autonomy architecture highlighting the overall flow of data between the subsystems. LiDAR points are ingested by DLIO [3] to produce an estimate of the robot's pose. The point cloud is both fed into the negative obstacle detector and processed by groundgrid [11] to segment non-ground points for the positive obstacle detector. Both detectors produces occupancy grids, which are combined to produce a risk map that is used by $A^\star$ to get a reference path. This reference path is optimized to be dynamically feasible and sent to the low-level controller.

2.1 Obstacle Detection

In order to safely navigate in outdoor unstructured environments, the UGV must be able to detect and avoid obstacles in its vicinity. We partition them into positive and negative obstacles. We note that positive obstacles are always dangerous, whereas risks are usually in the form of ***undetectable*** obstacles. All forms of obstacle detection occur within a local horizon of the UGV and are parsed into an occupancy grid centered at the location of the UGV with tunable height, width, and resolution H, W, and res. For this work, these are set to $(H, W, res) = (100\,\text{m}, 100\,\text{m}, 0.4\,\text{m})$.

Positive Obstacle Detection. The point cloud is processed by `groundgrid` [11], which segments the points into ground points and non-ground points. The non-ground points are projected onto the (x, y) plane and temporally averaged over the last M maps, with an occupancy threshold of $0.75M$. Then, the grid is dilated by 0.75m. In this implementation, M is set to 30.

Negative Obstacle Detection. Point cloud data is passed through a model-based detection system that searches for deviations in the ground points from expected geometries. It computes how much areas deviate from the expected nominal geometry, indicating the potential for risk, in this case, for negative obstacle detections. Nominally, for a ground plane, a LiDAR produces a series of concentric circles. Given the LiDAR mount height H_L and subsequent azimuth angles α_i and α_j, the expected distance between the scan points P_i and P_j can be calculated as Expected Distance$(P_i, P_j) = H_L / |\tan(\alpha_i) - \tan(\alpha_j)|$. Fig. 2 provides a visual of each of these quantities.

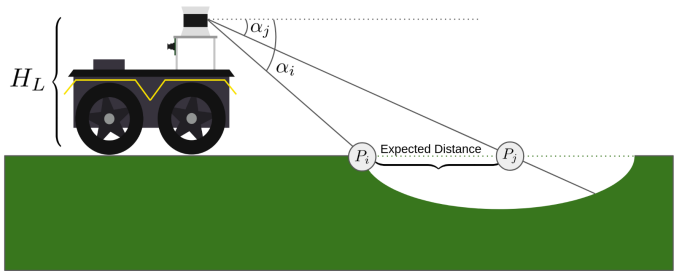

Fig. 2: H_L denotes the LiDAR mount height. Two scan lines are shown corresponding to azimuth angles α_i and α_j. Under nominal conditions when the UGV is on a flat plane, the scans are expected to return points P_i and P_j. In the presence of a negative obstacle, the distance between the two points will be larger than the expected distance.

Subsequent points in the LiDAR data that deviate from the expected model are labeled as corresponding feature points. For every corresponding pair, the risk metric is calculated as Risk Metric(P_i, P_j) = (Distance) / (Expected Distance). The Risk Metric is then used to populate a local occupancy grid, where this Risk Metric value is set for all cells spanned by the line segment between P_i and P_j. The last M' occupancy grids are averaged, such that the value at a given cell location is the median of the previous M' grids. M' is again set to 30 for this implementation. To produce smoother transitions between the boundaries of the identified negative obstacles and the surrounding environment, a Gaussian blur is applied.

Combined Risk Map Generation. The positive obstacle occupancy grid and the negative obstacle grid are merged to create a combined risk map to incorporate into a local 2-DOF UGV planner.

2.2 Path Planning & Control

Given a desired goal position, a coarse path is generated using the $A^\star$ algorithm. The path is passed into a 2-Degree of Freedom (DOF) optimization architecture. The first layer is an MPC optimization that generates a dynamically feasible reference trajectory. The reference trajectory is passed into the second layer, a smaller MPC optimization that generates control actions.

Coarse Path Generation. The coarse path is generated using $A^\star$. To allow for arbitrary planning horizons, a larger map is constructed to encompass both the UGV's starting position and the goal position. The map is updated locally using the combined risk maps. To account for risk in planning, the $A^\star$ cost between two edges u and v is formulated as $C(u, v) = c(u, v) \cdot (1 + \lambda M(v))$, where $c(u, v)$ is the euclidean distance, $M(v)$ is the risk value at v, and λ is a hyperparameter to modulate how risk-adverse the generated paths are. This formulation produces trajectories that attempt to remain short while circumventing high risk areas. The path is then made sparse using the Ramer-Douglas-Peucker algorithm, with

$\epsilon = 0.35$. Rectangular travel corridors are generated for each segment in the sparse path with a width of W_r and an additional height buffer of H_r on either end of the segment.

Dynamically Feasible Reference Trajectory Generation. The sparse $A^\star$ path and travel corridors are passed into the first layer of the optimization architecture. A local goal is chosen along the path, such that the goal is the last point on the path that remains within the local horizon of the combined risk map. The optimization is outlined in Eq. 1.

$$
\begin{aligned}
\min_{x[\cdot],u[\cdot]} \quad & \sum_{n=0}^{N-1} \left\{ \tilde{x}[n]^T R \tilde{x}[n] + u[n]^T Q u[n] \right\} \\
\text{s.t.} \quad & x_{n+1} = f(x[n], u[n]) && \text{(Dynamics Constraints)} \\
& x[n] \in P_j && \text{(Corridor Constraint)} \\
& x_{\min} \leq x[n] \leq x_{\max} && \text{(State Boundary Constraint)} \\
& a_{\min} \leq (u[n] - u[n-1])/dt \leq a_{\max} && \text{(Acceleration Constraint)} \\
& u_{\min} \leq u[n] \leq u_{\max} && \text{(Input Limit Constraint)} \\
& x[0] = \xi_s, u[0], u[n] = 0_{2\times 1} && \text{(Boundary Constraint)}
\end{aligned}
\tag{1}
$$

$\tilde{x}[n]$ denotes the error between the knot point and goal positions, and $u[n]$ denotes the control input at knot point n. f is the robot dynamics. P_j denotes a travel corridor. The travel corridor for knot point n corresponds to the travel corridor for the segment in the coarse path that encompasses $n/N\%$ of the total path length. The high-level MPC replans trajectories continuously as the UGV navigates towards the goal position to account for changes in the risk map.

Control. The reference trajectory is passed into the second layer of the optimization framework, which generates the next immediate control action. The second layer is outlined in Eq. 2.

$$
\begin{aligned}
\min_{x[\cdot],u[\cdot]} \quad & \sum_{n=0}^{N'-1} \left\{ (x[n] - x_{\text{ref}}[i+n])[n]^T R (x[n] - x_{\text{ref}}[i+n]) + u[n]^T Q u[n] \right\} \\
\text{s.t.} \quad & x_{\min} \leq x[n] \leq x_{\max} && \text{(State Boundary Constraint)} \\
& x[0] = \xi_s && \text{(Initial State Constraint)} \\
& u_{\min} \leq u[n] \leq u_{\max} && \text{(Input Limit Constraint)} \\
& a_{\min} \leq (u[n] - u[n-1])/dt \leq a_{\max} && \text{(Acceleration Constraint)} \\
& x_{n+1} = f(x[n], u[n]) && \text{(Dynamics Constraints)}
\end{aligned}
\tag{2}
$$

x_{ref} denotes the reference trajectory from the first layer. i is the index of the closest point in x_{ref} to the current position of the robot. N' is chosen to be much smaller than the original reference trajectory horizon. The first control value of the optimization is utilized as the control sent to the robot.

3 Experiments and Results

Hardware Platform The proposed framework is implemented on a Clearpath Jackal fitted with an Ouster OS1-64 LiDAR, which produces point clouds at a nominal rate of 10 Hz. It carries an onboard computer with an AMD Ryzen 5 3600 CPU, NVIDIA RTX 4000 Ada SFF GPU, and 32 GB of RAM.
Environments. We test our method in both a semi-urban office park and a rural environment. First, an evaluation of the negative obstacle detection is provided, where negative obstacle detections are compared to hand labeled data. Second, an evaluation of the proposed framework is conducted in which a UGV autonomously navigates to a goal in the presence of negative obstacles.

3.1 Negative Obstacle Detection Evaluation

The negative obstacle detector is evaluated in two different environments: a semi-urban office park and a rural road setting. In both environments, the generated risk maps are evaluated with ground truth labels, with the semi-urban office park using hand-labeled masks from orthomaps and the rural setting using hand-annotated point-cloud data.

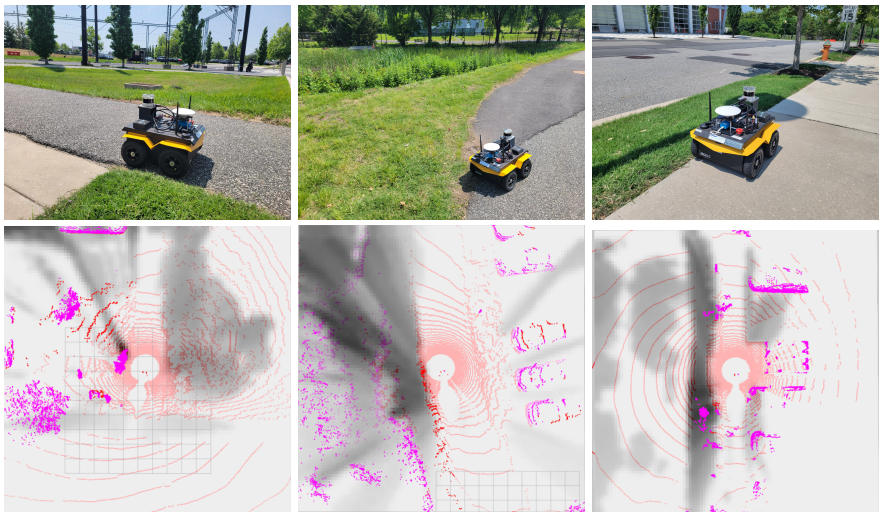

Fig. 3: Images of the validation sites: a small ditch, large ditch, and sidewalk curb. The top row shows the negative obstacles being detected, while the bottom row shows the generated risk maps, along with the segmented point cloud. Magenta corresponds to non-ground points. Darker cells indicate higher risk. Left: Small ditch (traversable but visually salient). Center: Large ditch (too steep for the UGV to exit). Right: Sidewalk curb (small but can get stuck).

Semi-Urban Office Park. There are three testing sites around the office park: a small ditch (traversable but visually salient), a large ditch (a steep slope pre-

Testing Site	IoU	Precision	Recall	F1
Small Ditch	0.4460	0.6960	0.5553	0.6164
Large Ditch	0.5697	0.7818	0.6803	0.7246
Curb	0.4348	0.7109	0.5174	0.5868

Table 1: Risk detection results from using the full risk map in the evaluation.

senting a navigation hazard), and sidewalk curbs (small abrupt recessions that can trap the UGV). For each location, five trials were conducted. For collected risk maps, negative obstacle masks were generated and compared against a hand-labeled mask derived from an orthomap of the test site. Sample images of the three test sites along with their corresponding detections are shown in Fig. 3. Around the small ditch, additional negative obstacles, such as a curb behind the UGV and segments of a curb to the front left, are also identified. The large ditch is strongly detected on the left of the UGV as well as portions of a parking lot curb on the right. The curb leading to a parking lot is only partially detected because portions of the curb are obstructed by cars in the area. For the curb experiments, the detector identifies borders on the left and corners on the right.

Table 1 reports the mean Intersection over Union (IoU), precision, recall, and F1 scores for each test site over a $20\,\mathrm{m} \times 20\,\mathrm{m}$ area, while Table 2 presents the same evaluation metrics computed over $10\,\mathrm{m} \times 10\,\mathrm{m}$ area. Cropping enables a more focused and precise assessment, as risk detection accuracy degrades as distance from the UGV increases. To better understand the failure modes in negative obstacle detection, Fig. 4 visualizes the detection labels for each test site. For the small ditch, only the steep, negative slopes are consistently detected. In the vicinity of the large ditch, a curb in the adjacent parking lot is partially occluded by positive obstacles, leading to incomplete detection. Additionally, the relatively small height difference between the curb and the road surface contributes to weak or missing negative obstacle labels. Some false positives are also observed, which can be attributed to the smoothing effects introduced by blurring the risk map.

Rural Environment. Point cloud data was collected along a road with ditches on either side. The point clouds were hand-annotated to identify ditch boundary points, which are then compared to feature points generated by the negative obstacle detector. The mean IoU, precision, recall, F1 score, and Chamfer distance were computed across LiDAR scans (Table 3). We emphasize the Chamfer distance metric, which measures closeness between point sets, as hand-annotated labels can introduce noise.

Testing Site	IoU	Precision	Recall	F1
Small Ditch	0.6313	0.7434	0.8183	0.7716
Large Ditch	0.4773	0.5436	0.8016	0.6434
Curb	0.4983	0.6416	0.6886	0.6620

Table 2: Risk detection results from using the cropped risk map in the evaluation.

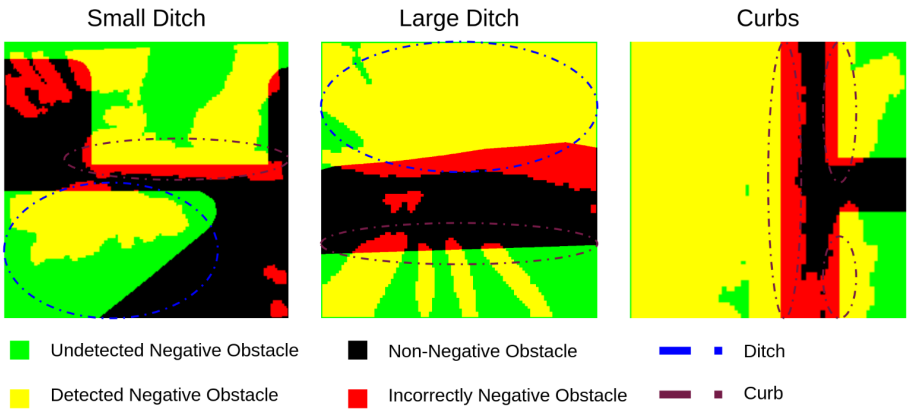

Fig. 4: From left to right, the images show the detection labels of the small ditch, large ditch, and the curb. Green is ground truth negative obstacles, and black is ground truth safe terrain. Yellow indicates a correct labeling as a negative obstacle and red indicates incorrect labeling as a negative obstacle.

Mean Metric Across 20 LiDAR Scans

IoU	**Precision**	**Recall**	**F1**	**Chamfer Distance**
0.472	0.569	0.742	0.652	0.177

Table 3: Risk detection results on collected data from the rural environment.

3.2 Risk-averse Navigation

The UGV was commanded to navigate to goal positions near negative obstacles at several semi-urban office park locations. The desired behavior was to either circumvent negative obstacles or descend them in a risk-averse manner. Testing sites included the small ditch, large ditch, a hill with sharp drop-offs, a sharp descent in a park, and a long hill.

Large Ditch. The large ditch highlighted a challenge for the risk detection: occlusion. At the start of the experiment, tall grass obscured the ditch, preventing early detection. As the UGV advanced alongside the ditch, portions of the negative obstacle gradually came into view. Once the ditch came into view, the generated plans began to adjust around areas with risk.

Small Ditch. This experiment revealed a limitation with the negative obstacle detector: since the system detects negative slopes, positive slopes (like the ditch's far edge) aren't identified as risks. Only the near edge was detected. In flatter sections with low perceived risk, the UGV entered the ditch and became level with terrain, no longer perceiving it as a negative obstacle.

Hill with Sharp Drop-Offs. The UGV successfully avoided a steep left-side slope and chose a gradual descent path toward the goal, demonstrating effective high-risk terrain identification and avoidance.

Sharp Descent in Park. This scenario revealed a positive obstacle detection failure mode. Starting at a higher altitude than the goal, foliage level with the UGV marked the goal location as occupied, preventing the planner from generating a plan, despite valid paths existing. This highlights needs for better terrain semantic integration and planning framework flexibility.
Long Hill. The negative obstacle detector performed well, with planned paths tracing the hill boundary and avoiding drop-offs effectively. However, another limitation emerged for positive obstacle detection. Rapid UGV movement caused failures in detections from partial views. Combined with finite $A^\star$ replanning rates, this resulted in collisions, emphasizing needs for improved planning frequencies and mapping capabilities.

4 Lessons Learned & Future Work

Our experiments revealed three main failure modes in the current system: odometry drift, sensitive positive obstacle detection, and slow planning rates. These limitations collectively impacted the robot's ability to navigate safely and consistently in unstructured environments. Below, we discuss each issue in detail and outline corresponding improvements for future work.
Noisy Odometry. The most significant source of error was odometry drift as it drifted when the robot encountered vibrations and rapid changes in elevation. This caused the estimated goal position to drift into unreachable or invalid areas, such as inside positive or negative obstacles. In extreme cases, the drift placed the robot outside the bounds of the global map, rendering planning infeasible. Future work will explore more robust odometry solutions that integrate inertial sensing and terrain-aware corrections to maintain localization in rough conditions.
Positive Obstacle Detection Methodology. The current positive obstacle detection pipeline proved sensitive in dynamic and uneven environments. At high speeds, many obstacles were missed due to temporal gaps in grid updates. Moreover, uneven terrain caused obstacles to shift rapidly in the grid frame, often disappearing before being registered as hazards. The approach struggles with elevated obstacles relative to the ground, such as overhanging branches, that may be at the same global height as the robot, but do not pose a risk as the ground is much lower. To address this, we propose transitioning away from segmented point cloud approaches and toward a column-wise analysis of vertical LiDAR slices. By estimating the local ground height per column and classifying obstacles relative to this baseline, the system can more reliably detect both near-ground and elevated hazards.
Replanning Frequency. The global planner, based on $A^\star$, showed performance bottlenecks when operating at scale. In large environments, especially those with frequent changes in perceived risk, the planner struggled to update paths in real time. This issue was further amplified by noise in the obstacle map caused by odometry drift and poor segmentation—particularly in grassy areas—leading the robot to believe it was stuck in an obstacle. Future work will focus on improving the efficiency of the implementation.

Acknowledgements

We gratefully acknowledge the support of ARL DCIST CRA W911NF-17-2-0181. This work was also partially founded bythe IoT4Ag Engineering Research Center funded by the U.S. National Science Foundation (NSF) under NSF Cooperative Agreement Number EEC-1941529, and NVIDIA.

References

[1] Paulo Borges et al. "A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges". In: *Field Robot* 2.1 (2022), pp. 1567–1627.

[2] Xiaoyi Cai et al. "EVORA: Deep EVidential Traversability Learning for Risk-Aware Off-Road Autonomy". In: *arXiv preprint arXiv:2311.06234* (2023).

[3] Kenny Chen, Ryan Nemiroff, and Brett T Lopez. "Direct LiDAR-Inertial Odometry: Lightweight LIO with Continuous-Time Motion Correction". In: *2023 IEEE Int. Conference on Robotics and Automation (ICRA)* (2023), pp. 3983–3989. DOI: 10.1109/ICRA48891.2023.10160508.

[4] Anushri Dixit et al. "STEP: Stochastic Traversability Evaluation and Planning for Risk-Aware Navigation; Results From the DARPA Subterranean Challenge". In: *IEEE Transactions on Field Robotics* 2 (2025), pp. 81–99. DOI: 10.1109/TFR.2024.3512433.

[5] Jonas Frey et al. "Fast Traversability Estimation for Wild Visual Navigation". In: *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea, July 2023. DOI: 10.15607/RSS.2023.XIX.054.

[6] Jonas Frey et al. "Roadrunner-learning traversability estimation for autonomous off-road driving". In: *IEEE Trans. on Field Robotics* (2024).

[7] Xiangyun Meng et al. "TerrainNet: Visual Modeling of Complex Terrain for High-speed, Off-road Navigation". In: Mar. 2023.

[8] Ian D Miller et al. "Air-Ground Collaboration with SPOMP: Semantic Panoramic Online Mapping and Planning". In: *IEEE Trans. on Field Robotics* (2024).

[9] Simone Palazzo et al. "Domain Adaptation for Outdoor Robot Traversability Estimation from RGB data with Safety-Preserving Loss". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 10014–10021. DOI: 10.1109/IROS45743.2020.9341044.

[10] Amirreza Shaban et al. "Semantic Terrain Classification for Off-Road Autonomous Driving". In: *Proceedings of the 5th Conference on Robot Learning*. Ed. by Aleksandra Faust, David Hsu, and Gerhard Neumann. Vol. 164. Proceedings of Machine Learning Research. PMLR, Nov. 2022, pp. 619–629. URL: https://proceedings.mlr.press/v164/shaban22a.html.

[11] Nicolai Steinke, Daniel Goehring, and Raúl Rojas. "GroundGrid: LiDAR Point Cloud Ground Segmentation and Terrain Estimation". In: *IEEE Robotics and Automation Letters* 9.1 (2024), pp. 420–426. DOI: 10.1109/LRA.2023.3333233.