The Journal of Open Source Software

# traversability_generator3d: A C++ library for 3D traversability estimation from MLS maps

**Muhammad Haider Khan Lodhi**[1], **Christoph Hertzberg**[1], **Arne Böckmann**[2], **and Janosch Machowinski**[2]

**1** Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI), Bremen, Germany **2** Cellumation GmbH

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:**

**Submitted:** 15 September 2025
**Published:** unpublished

## Summary

`traversability_generator3d` is a C++ library that converts Multi-Level Surface (MLS) (Triebel et al., 2006) maps into 3D traversability maps. Unlike traditional 2D elevation-based maps, MLS preserves multiple surface hypotheses per cell, making it well-suited for environments with rubble, vegetation, bridges, or overlapping terrain layers. The library builds on the MLS representation to provide robot-specific traversability analysis, where each grid cell is enriched with slope, step height, plane fitting results, allowed motion orientations, and optional soil semantics. The result is a `TraversabilityMap3d` suitable for autonomous navigation. It is designed for real-world robotic applications including construction robotics, disaster response, search-and-rescue, and planetary exploration, where terrain is uneven, uncertain, and multi-layered.

`traversability_generator3d` was originally developed as part of ugv_nav4d (Böckmann et al., 2024) and later separated into a stand-alone library to facilitate independent use and broader applicability.

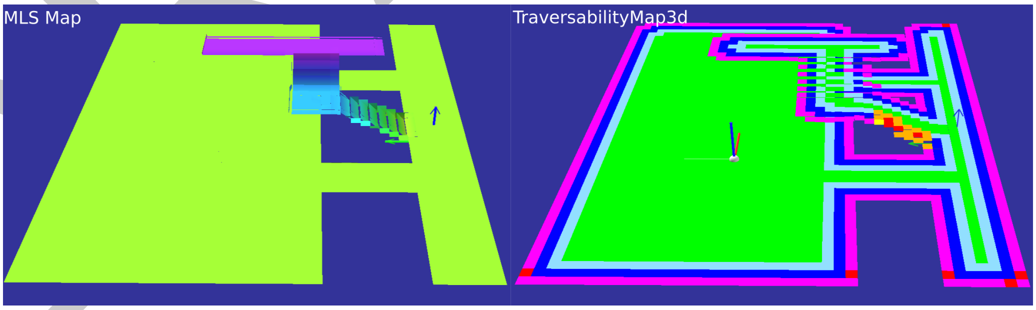

**Figure 1:** Comparison between the original Multi-Level Surface (MLS) map (left) and the generated TraversabilityMap3d (right). The MLS encodes the underlying 3D terrain structure, whereas the TraversabilityMap3d highlights navigable and non-navigable regions for path planning.

## Statement of need

Safe and efficient navigation requires robots to distinguish where they can move from where they cannot. This is especially critical for outdoor field robots, walking excavators, and planetary rovers that face slopes, steps, loose soil, or debris. Existing approaches often use 2.5D elevation maps (Fankhauser et al., 2018) that store only one surface per XY cell, losing critical information when multiple layers overlap, voxel/octree maps (Hornung et al., 2013) preserve 3D structure but are memory-intensive and less tailored for traversability, and purely semantic approaches (Cramer et al., 2018; Wiesmann et al., 2021) that classify terrain appearance but often neglect physical constraints like slope or robot footprint.

`traversability_generator3d` addresses these limitations by:

- Taking MLS as input, retaining multi-level terrain structure.
- Performing plane fitting and slope analysis per patch, allowing orientation-dependent motion constraints.
- Checking step height and body collisions with robot-specific geometry (AABB/OBB checks).
- Supporting soil-aware traversability, where semantic labels (sand, gravel, rocks, concrete) are integrated with uncertainty.
- Producing a 3D traversability map with rich node types that planners can use directly.

## Input

The library consumes Multi-Level Surface (MLS) maps, which store multiple surface hypotheses per grid cell. This allows the library to reason about environments with overlapping structures (e.g. bridges, rubble, vegetation), unlike simpler 2.5D elevation maps.

## Processing pipeline

The traversability generation proceeds in several stages:

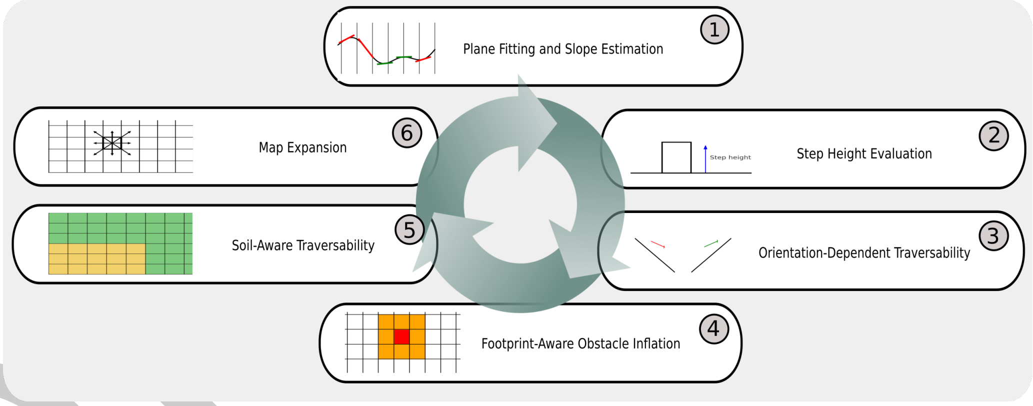
Plane Fitting and Slope Estimation
1
Map Expansion
6
Step height
Step Height Evaluation
2
Soil-Aware Traversability
5
Orientation-Dependent Traversability
3
Footprint-Aware Obstacle Inflation
4

**Figure 2:** Traversability generation pipeline showing plane fitting, step height checks, orientation limits, footprint inflation, soil integration, and map expansion.

1. Plane fitting and slope estimation
    - Local surface patches are extracted from the MLS and fitted with planes using RANSAC (via PCL).
    - Each patch's slope angle and slope direction are computed from the fitted plane normal.
2. Step height evaluation
    - Both axis-aligned (AABB) and oriented bounding box (OBB) checks are performed to test whether nearby patches exceed the robot's configured maximum step height.
    - This ensures that steps or discontinuities larger than the robot's physical capability are marked as obstacles.
3. Orientation-dependent traversability
    - On steep slopes, the library restricts the allowed orientations of motion (incline limiting).

   - For example, forward uphill/downhill motion may be permitted.

4. Footprint-aware obstacle inflation
   - Obstacles and frontiers are grown outward by the robot's footprint radius.
   - This guarantees that planned paths account for the full robot body, not just its center point.
5. Soil-aware traversability
   - In addition to geometry, the library maintains a parallel `TraversabilityMap3d<SoilNode*>` to model soil types.
   - Soil samples (sand, gravel, rocks, concrete) can be injected with associated uncertainty.
   - A Gaussian propagation mechanism distributes soil probabilities spatially, and costs are updated accordingly.
   - Certain soils may be forbidden entirely (e.g., robots not allowed to enter sand), in which case nodes are automatically converted to obstacles.
6. Map expansion
   - The library grows the traversability map outward from user-defined start nodes.
   - Unknown areas are classified as `FRONTIER` nodes, guiding exploration.
   - Expansion can be bounded by distance to restrict computation to a region of interest.

## Output

The result is a `TraversabilityMap3d` structure with two layers:

- Geometric traversability (`TravGenNode`) containing slope, plane, step height, allowed orientations, cost, and node type.
- Soil traversability (`SoilNode`) containing soil type probabilities and semantic costs.

# Traversability node types

Each cell in the `TraversabilityMap3d` is represented as a `TravGenNode` with geometric, semantic, and cost attributes.
Nodes are categorized as:

- `TRAVERSABLE`: safe patches where the robot can move.
- `OBSTACLE`: unsafe regions due to excessive slope, step height, body collisions, or forbidden soil.
- `FRONTIER`: expansion boundaries between known and unknown areas.
- `INFLATED_OBSTACLE`: obstacles grown outward to include robot footprint.
- `INFLATED_FRONTIER`: expanded frontiers with safety margins.
- `UNKNOWN`: regions with insufficient or missing MLS data.
- `HOLE`: gaps where too few patches exist to support safe navigation.
- `UNSET`: unclassified nodes during initialization.

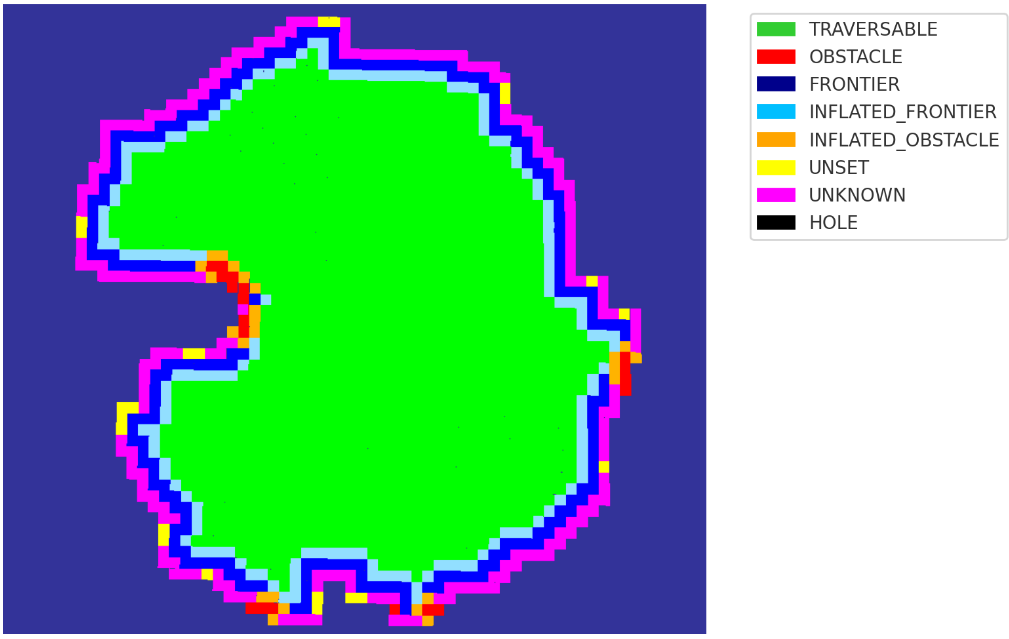

**Figure 3:** Traversability map with semantic node classifications.

This classification enables downstream planners to reason not only about safe/unsafe areas but also about uncertainty and exploration frontiers.

## Soil-aware traversability

In addition to geometry, `traversability_generator3d` models soil composition. Each `SoilNode` stores probability distributions over soil types. By default, costs are adapted as follows:

- `Concrete`: low cost (preferred).
- `Sand/Gravel`: higher cost (risky terrain).
- `Rocks`: costly or forbidden depending on configuration.
- `Unknown soil`: assigned maximum cost.

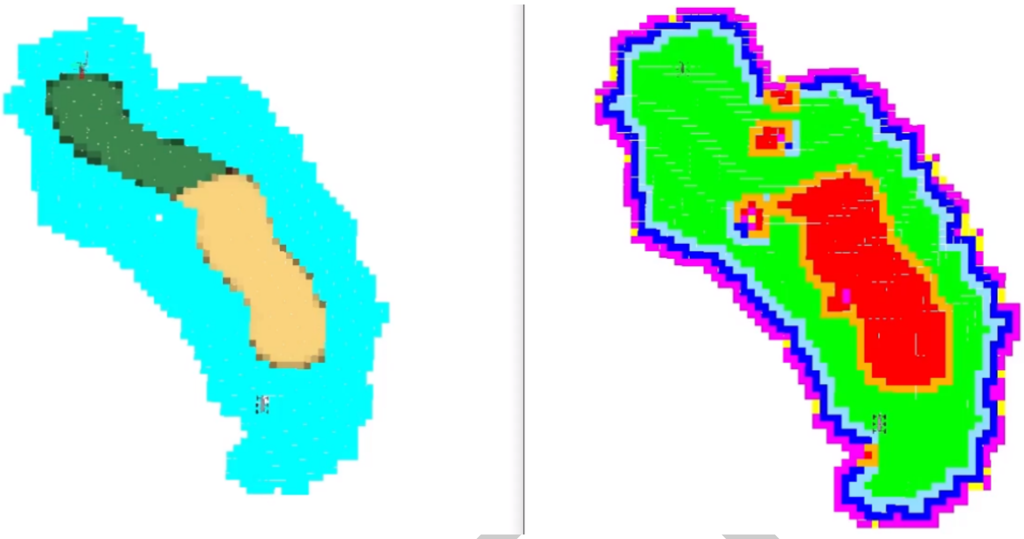

**Figure 4:** Map layers for terrain assessment. Left: Soil map with sand (yellow), gravel (green), and unknown (light blue). Right: Corresponding traversability map, where sand is treated as obstacle (red).

Through configuration, users can forbid traversal on certain soils, automatically converting affected nodes to obstacles. This enables integration of perceptual information (e.g. from ground-penetrating radar, visual classifiers, or tactile sensors) with geometric terrain reasoning.

## Field Tests

The traversability_generator3d library has been used extensively in research projects for almost a decade. It has supported autonomous navigation in several projects, including Entern (Schwendner et al., 2015), ANT (Dettmann et al., 2022), VIPE (Dettmann et al., 2018), KiMMI-SF, HiSE, PerSim, CoRob-X (Domínguez et al., 2025), Robdekon2, and NoStrandAMust.

As a versatile component for terrain assessment, it has also been applied on a wide range of terrestrial robots at the DFKI Robotics Innovation Center (RIC), such as Arter, Artemis, Coyote-III, Crex, Charlie, Hunter-SE, SherpaTT, and Asguard-IV.

## Acknowledgements

The traversability_generator3d library was initiated and is being developed at the Robotics Innovation Center of the German Research Center for Artificial Intelligence (DFKI) in Bremen, in close collaboration with the Robotics Group of the University of Bremen. Its development began within the framework of the Entern project (grant no. 50RA1406) and has since been further advanced in the context of Robdekon2 (grant no. 13N16537) and NoStrandAMust (grant no. 50RA2122). The work has been supported by the German Aerospace Center (DLR) with funding from the German Federal Ministry for Economic Affairs and Climate Action (BMWK) as well as by the German Federal Ministry of Education and Research (BMBF).

We would also like to acknowledge the contributors of the traversability_generator3d repository, including the developers visible on the project's GitHub contributors page, for their valuable efforts and dedication.

## References

Böckmann, A., Machowinski, J., & Lodhi, M. H. K. (2024). ugv_nav4d: Advanced Multi-Surface Navigation for Unmanned Ground Vehicles Using 4D Path Planning Techniques. *Journal of Open Source Software*, *9*(104), 6983. https://doi.org/10.21105/joss.06983

Cramer, D., Schulz, H., & Behnke, S. (2018). Traversability analysis for mobile robots in outdoor environments: A survey. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7624–7631.

Dettmann, A., Kuehn, D., Van Opdenbosch, D., Stark, T., Peters, H., Natarajan, S., Kasperski, S., Boeckmann, A., Garcea, A., Steinbach, E., & others. (2018). Exploration in inaccessible terrain using visual and proprioceptive data. *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Madrid, Spain*.

Dettmann, A., Planthaber, S., Bargsten, V., Dominguez, R., Cerilli, G., Marchitto, M., Fink, G., Focchi, M., Barasuol, V., Semini, C., & others. (2022). Towards a generic navigation and locomotion control system for legged space exploration. *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2022)*.

Domínguez, R., Pérez-del-Pulgar, C., Paz-Delgado, G. J., Polisano, F., Babel, J., Germa, T., Dragomir, I., Ciarletti, V., Berthet, A.-C., Danter, L. C., & others. (2025). Cooperative robotic exploration of a planetary skylight surface and lava cave. *Science Robotics*, *10*(105), eadj9699.

Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, M., Hutter, M., & Siegwart, R. (2018). Probabilistic terrain mapping for mobile robots with uncertain localization. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1875–1882.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, *34*(3), 189–206.

Schwendner, J., Carrió, J., Domínguez, R., Plan-thaber, S., Yoo, Y.-H., Asadi, B., Machowinski, J., Rauch, C., & Kirchner, F. (2015). Entern: Environment modelling and navigation for robotic space-exploration. *Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA),(ASTRA), Noordwijk, in Proceedings ASTRA*.

Triebel, R., Pfaff, P., & Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2276–2282.

Wiesmann, L., Furrer, F., Siegwart, R., & Cadena, C. (2021). Traversability estimation for mobile robots in natural environments: A survey. *Frontiers in Robotics and AI*, *8*.