

Comparative Evaluation of Navigation Meshes for Martian Rover Operations

Karol Orzechowski, Piotr Palka
Warsaw University of Technology

September 11, 2026

Abstract

Autonomous navigation is a critical capability for future Mars exploration missions, where communication delays, the absence of a global positioning system, and challenging terrain conditions impose stringent requirements on rover navigation systems. This study investigates the applicability of navigation meshes for representing traversable Martian terrain and supporting autonomous path planning. A Martian analog research field at the Warsaw University of Technology in Józefosław near Warsaw was reconstructed from high-resolution LiDAR point-cloud data and used as a controlled experimental environment. Three navigation mesh approaches—Recast and Detour, NeoGen, and Explicit Corridor Map—were considered, with Recast and Detour and NeoGen evaluated using a set of metrics covering terrain coverage, geometric accuracy, mesh complexity, memory requirements, construction performance, and data compression. The results reveal substantial differences between the evaluated approaches. Recast and Detour produced a significantly more compact and computationally efficient navigation mesh, while NeoGen provided higher terrain coverage and a better compression ratio at the cost of increased mesh complexity, computational requirements, and mapping error. The findings indicate that Recast and Detour is currently more suitable for resource-constrained Martian rover navigation, whereas NeoGen shows potential in applications where detailed terrain representation and efficient data transmission are prioritized. The study also highlights important challenges related to terrain resolution, mesh validation, scalability, and implementation efficiency. These results provide a foundation for further research on adaptive navigation meshes, multi-agent rover navigation, and autonomous path planning in Martian environments.

1 Introduction

Navigation on Mars presents unique challenges compared to Earth-based scenarios. The significant communication delay between Earth and Mars (ranging from 4 to 24 minutes) [11, 4] necessitates substantial autonomy in navigation decisions. Unlike Earth, Mars lacks a global positioning system, forcing rovers to rely on visual odometry and terrain feature recognition for localization [10, 2]. Furthermore, the Martian terrain presents exceptional challenges with its varied surface features including steep slopes, loose soil, and rocky outcrops that can trap or damage rovers [6, 15]. These constraints make robust navigation algorithms particularly crucial for successful Mars exploration missions.

As navigation is an unavoidable part of Martian exploration, it involves moving multiple rovers – hereafter referred to as agents [3] – through the complex environment of Mars’ surface. While the Martian terrain is mostly static, it can occasionally change due to weather conditions, and errors in terrain mapping may occur. This necessitates creating adaptive maps that can adjust to these changes [19]. Such maps identify accessible areas of the surface and enable experimentation with specific rovers.

A navigation mesh [18] is a data structure that describes the accessible parts of a terrain for a specific agent type. Typically, it consists of a set of regions that are later utilized in pathfinding algorithms.

Navigation meshes are well-known structures in the world of computer games, where they define areas where virtual objects or bots can move within an environment [8]. Modern implementations support multiple characters or bots, enabling them to navigate to specific points along chosen paths. Essentially, navigation meshes are crucial for facilitating pathfinding.

In robotics, similar concepts exist under the term meadow maps [1]. Insights gained from these applications can also be applied to space exploration, particularly for navigating the surface of Mars.

In case of this study, navigation meshes supporting multiple agents will be used to enable pathfinding. These agents may represent multiple autonomous rovers operating simultaneously within the same environment. This multi-agent setup reflects realistic scenarios of Martian exploration, where several robotic units may need to coordinate their movements, distribute tasks, or respond dynamically to environmental changes and mission goals.

Given these specific conditions, this work aims to test how different navigation mesh generation algorithms would perform if applied to the surface of Mars. To evaluate the performance, metrics based on coverage, connectivity, complexity, and computation performance are introduced following methodologies described in [18] and [13]. Additionally, the reconstruction of the Professor Mieczysław Bekker Martian analog research field, belonging to the Warsaw University of Technology and located in Józefosław near Warsaw, is outlined, demonstrating key preprocessing steps to derive navigation meshes.

2 Related work

Navigation mesh research has been driven by several independent groups, and the methods compared in this paper come from different ones. Kallmann and Kapadia [8] surveyed the family as a whole and placed it in the context of real-time dynamic planning for virtual worlds, establishing the vocabulary of cells, portals and free space that the later work shares.

At Utrecht University, Van Toll, Cook and Geraerts [19] introduced a navigation mesh that can be updated while the environment changes, so that moving obstacles and terrain modifications are absorbed by a local repair rather than by a full rebuild. The same group later formalised the explicit corridor map [16], in which Van Toll, Cook, Van Kreveld and Geraerts represent the navigable space by its medial axis and the associated clearance, giving paths with an adjustable margin from the obstacles instead of a fixed decomposition into regions.

At the Universitat Politècnica de Catalunya, Oliva and Pelechano [13] took a different route with NEOGEN, which recovers a near-optimal convex decomposition of multi-layered three-dimensional environments directly from their geometry. Their emphasis is on the quality of the decomposition itself, that is, on how few and how well-shaped the convex cells are, rather than on the auxiliary structures built over it.

These lines of work were brought together in the comparative study of Van Toll, Triesscheijn, Kallmann, Oliva, Pelechano, Pettré and Geraerts [18], a collaboration between the authors of the competing methods. It is the source of the evaluation framework used here, grouping the properties of a navigation mesh into coverage, connectivity, convexity, adjustment, complexity and performance, and it makes the point that these groups trade against one another rather than improving together.

Outside academia, the Recast and Detour toolset of Mononen [12] became the de facto standard in the games industry. It has no accompanying publication, but its voxelisation-based pipeline is widely reimplemented and is one of the two generators evaluated below.

The idea predates the term. In robotics, Arkin [1] described meadow maps, a decomposition of the free space of a vision-based mobile robot into convex regions, which is the same construction under a different name. On the data side, Wack and Wimmer [20] developed grid-based digital terrain models from airborne laser scanner data, the step that connects a raw survey of real terrain to any of the mesh generators above, and the one this work depends on to turn a point cloud of the research field into an input surface.

3 Definitions

Navigation meshes and their application in Martian exploration involve several specialized concepts and components [18]. These elements work together to create an effective navigation system that can handle the unique challenges of extraterrestrial terrain [11]. The integration of these components requires precise definitions to ensure consistent implementation and evaluation across different scenarios. The following subsections define the key terms and concepts used throughout this paper.

3.1 Agent

An agent represents a rover operating within the Martian environment. Each agent is modeled with specific constraints, such as size, battery capacity, and turning radius [3]. For this study, the LeoRover platform [9] is used as a representative agent. There will be multiple heterogenic agents, meaning agents with differing capabilities or functionalities. This diversity will enhance possibilities of Mars exploration.

LeoRover is an open-source, four-wheeled mobile robot platform designed for research and development purposes. The values below follow the manufacturer’s technical specification, revision 1.9 [9], and are the values that drive the navigation mesh parameters in Section 5.1:

- **Physical Dimensions:** 424 x 445 x 303 mm (L x W x H)
- **Weight:** 7 kg (without additional payload)
- **Maximum Payload:** 5 kg
- **Maximum Speed:** 0.4 m/s
- **Turning Radius:** 0 m (can rotate in place)
- **Maximum Obstacle Height:** 70 mm
- **Climbing Ability:** Up to 45° incline

These specifications make LeoRover suitable for simulating Martian rover navigation scenarios, as it shares similar mobility constraints and operational parameters with actual planetary rovers, albeit at a smaller scale. The platform’s ability to handle uneven terrain and its robust construction make it particularly appropriate for testing navigation mesh algorithms in analog environments. Two of these numbers matter more than the rest for what follows: the width of 445 mm, which fixes the clearance radius of 0.2225 m used by both generators, and the maximum obstacle height of 70 mm, which fixes the maximum step the classification may treat as drivable.

3.2 Complex Terrain

Complex terrain refers to an environment featuring varied elevation, obstacles, and regions of limited accessibility [1]. In this study, the complex terrain is modeled after a reconstruction of the Professor Mieczysław Bekker Martian analog research field, shown in Figure 1. The field was established at CENAGIS (Centre for Scientific Geospatial Analyses and Satellite Computations) of the Warsaw University of Technology’s Faculty of Geodesy and Cartography, in Józefosław near Warsaw [14], and was specifically designed to simulate Martian terrain conditions. It carries the name of Mieczysław Bekker, the Polish engineer who founded terramechanics, the study of vehicle locomotion over deformable ground, and who co-designed the Lunar Roving Vehicle. The name is apt for the work reported here, because what the field reproduces is precisely the interaction between a wheeled vehicle and loose, uneven terrain. The short form *the Bekker field* is used in the remainder of this paper.

3.3 Drivable Terrain

Traversable section of complex terrain is called ”drivable terrain”. It consists of areas that are traversable by the agent. These areas are determined based on slope, surface material, and obstacles [19]. An area of 10 × 10 meters was used, filled with crushed stone of various grain sizes, and terrain features characteristic of Mars were formed (a crater, ripples, and loosely scattered larger rocks). The finest fraction was omitted.

3.4 Navigation Mesh

A navigation mesh is a geometric representation of drivable terrain, subdivided into connected regions [8]. Each region facilitates pathfinding by providing adjacency information for movement. The algorithms below are being considered for analysis because they represent different navigation mesh generation techniques, some of which are well-known in the gaming industry.



Figure 1: Research field in Józefosław. The field incorporates several key features designed to simulate Martian terrain: a single primary crater formation, four short ripples, one long forked ripple structure, large stones strategically positioned throughout the field, and multiple zones with varying stone granularity.

- **Recast and Detour** – A widely used navigation mesh generation algorithm[12] that utilizes voxelization to divide the terrain into walkable regions. It is particularly effective for large-scale environments due to its robustness and scalability. Detour, a companion library, provides pathfinding and spatial queries on the generated mesh. This approach is commonly applied in robotics and video games, offering efficient navigation in dynamic environments.
- **NeoGen** – A navigation mesh generation technique[13] designed for complex, multi-level environments. It focuses on handling vertical connectivity and irregular terrain features, making it suitable for environments with significant elevation changes, such as urban structures or rough Martian terrain. NeoGen emphasizes accurate terrain representation and improved pathfinding precision in challenging conditions. The implementation used here follows the single-layer case of that pipeline: the terrain is sampled into a height grid, each grid cell is classified as drivable or not from the local slope and the step height to its neighbours, the drivable cells are grouped into connected regions, the contour of the selected region and of every obstacle inside it is traced and simplified, the resulting polygon with holes is triangulated under those constraints, and the triangles are merged greedily into convex cells. Cells that share an edge are linked by a portal, so the output is not only a set of polygons but an adjacency graph over them.
- **Explicit Corridor Map** – This method represents[16] navigable space as a set of corridors rather than discrete regions. It provides greater flexibility in motion planning by allowing agents to traverse paths with minimal deviation and improved obstacle avoidance. ECM is particularly useful in scenarios requiring precise movement control and efficient path optimization.

4 Reconstruction of the Bekker Research Field in Józefosław

The field features various surface types including loose soil, rocks of different sizes, and carefully crafted slopes that mimic the challenging terrain encountered on Mars [20].

Complex terrain was reconstructed using advanced terrain processing techniques, involving the following steps:

1. **Drone Scanning:** A high-resolution LiDAR (velodyne puck VLP-16) scan was conducted using a drone. This generated a detailed point cloud of the research field, capturing all relevant terrain features including the artificial craters and various surface obstacles that were deliberately placed to simulate Martian conditions.
2. **Height Map Generation:** From the point cloud, a height map was derived. This involved projecting the points onto a 2D plane and calculating elevation values for each grid cell, accurately representing the field’s varied topography, which includes slopes ranging from 15 to 30 degrees.
3. **Field Boundary Delineation:** Boundaries of the research field were identified using the initial scan. This step included manual or semi-automatic annotation of relevant edges to isolate the test area, which was specifically constructed to contain distinct zones with different levels of difficulty for rover navigation.
4. **Point Cloud Cropping:** The point cloud was cropped to match the delineated field boundaries. This ensured that only the points within the defined area were retained for further processing.
5. **Mesh Construction:** Using a grid-based approach, the filtered point cloud was converted into a navigation mesh. This included:
 - *Point Filtering:* Removing noise and outliers from the point cloud to enhance accuracy.
 - *Grid Generation:* Dividing the terrain into a uniform grid and interpolating elevation values where necessary.
 - *Mesh Generation:* Constructing a mesh based on the grid, with additional refinement steps to ensure smooth transitions and optimal connectivity.

The reconstructed field, which can be seen in Figure 2 serves as an ideal testing ground for navigation algorithms, as it incorporates features specifically designed to challenge rover mobility and navigation systems [14]. The terrain includes various obstacles and surface types that are representative of actual Martian surface conditions, making it particularly valuable for validating navigation mesh generation algorithms in a controlled yet realistic environment.

This reconstruction process highlights the importance of preprocessing in creating reliable navigation meshes for real-world applications, particularly in the context of planetary exploration simulation.

4.1 Scale of the reconstructed model

The reconstruction pipeline does not emit metres. Point coordinates are converted to millimetres before gridding, and the reorientation step that puts the height on the Y axis rescales the model by a further factor of 0.01, so one unit of the stored model corresponds to 0.1 m. Left unstated, this is exactly the kind of discrepancy that makes area figures uninterpretable and lets agent parameters be a factor of ten off. All results reported here are therefore computed on a copy of the model rescaled by 0.1, in which one unit is one metre. The model then spans 14.59×15.06 m, its elevation varies by 0.79 m, and the drivable surface derived from it measures about 100 m² out of the 220 m² the scan covers, consistent with the 10×10 m field surrounded by the margin the scan happened to capture. Every parameter and every area in the remainder of this paper is expressed in that metric frame.

5 Methodology

The experiments were conducted on the reconstructed research field in Józefosław to simulate Martian conditions. The following steps outline the experimental procedure, shown also on diagram 3.

1. **Scenario Design:** In this paper only one test scenario is used – the original shape of the research field.
2. **Algorithm Comparison:** Three algorithms were tested:
 - Recast and Detour [12]
 - NeoGen [13]

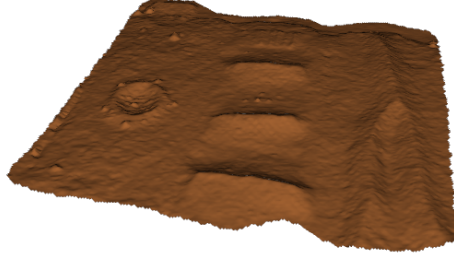


Figure 2: Reconstructed 3D model of the Martian analog research field in Józefosław, generated from LiDAR point cloud data. The visualization shows the detailed terrain features including artificial craters, slopes, and surface variations designed to simulate Martian surface conditions. The model serves as the base input for navigation mesh generation algorithms.

- Explicit Corridor Map [16]

3. **Metrics Evaluation:** The evaluation framework incorporates metric groups drawing from established methodologies in navigation mesh analysis in [18] and [13]. Selected metrics from each group are detailed in the subsequent Metrics section.

5.1 Experimental Parameters

Both generators are driven from one shared parameter file and run on the same metric terrain model (Section 4.1), so every difference reported in Section 6 is a property of the algorithm rather than of its settings. This is worth stating explicitly, because the parameters that matter most here are expressed in metres. Feeding one generator a terrain at a different scale than the other, while giving both the same nominal maximum step height, would let one of them climb a step ten times larger relative to the ground it is classifying, and no amount of care in the metrics afterwards would repair that.

The agent parameters are taken directly from the LeoRover specification of Section 3.1: a clearance radius of 0.2225 m (half the 445 mm width), an agent height of 0.303 m and a maximum step of 0.07 m. The one place where this study deliberately departs from the manufacturer’s numbers is the slope limit. LeoRover is rated for inclines up to 45°, but at that threshold the drivable/non-drivable classification on this particular reconstruction bridges straight across the rim of the main crater: the rim is noisy enough that individual facets fall below 45°, and the flood fill walks through the gaps into terrain that is in fact inaccessible. At 20° both generators wall the crater off cleanly and agree on the total drivable area to within a few percent, so 20° is used throughout. The limitation is one of the reconstruction rather than of the platform: the rated 45° is a statement about the rover on a compacted surface, not about what can be recovered reliably from a noisy point cloud of loose gravel.

The cell size of 0.05 m is roughly a quarter of the agent radius, comfortably inside the range Recast recommends (radius/2 to radius/3) and fine enough to resolve the individual stones placed on the field.

The minimum obstacle area deserves a note, because it is the one threshold that materially changes what NeoGen returns. Holes below roughly the agent’s own footprint are treated as classification noise rather than as rock, which is reasonable on a decimated scan where a two-cell steep patch is usually slope flicker. The implementation additionally carried an absolute floor of 1 m² on that threshold,

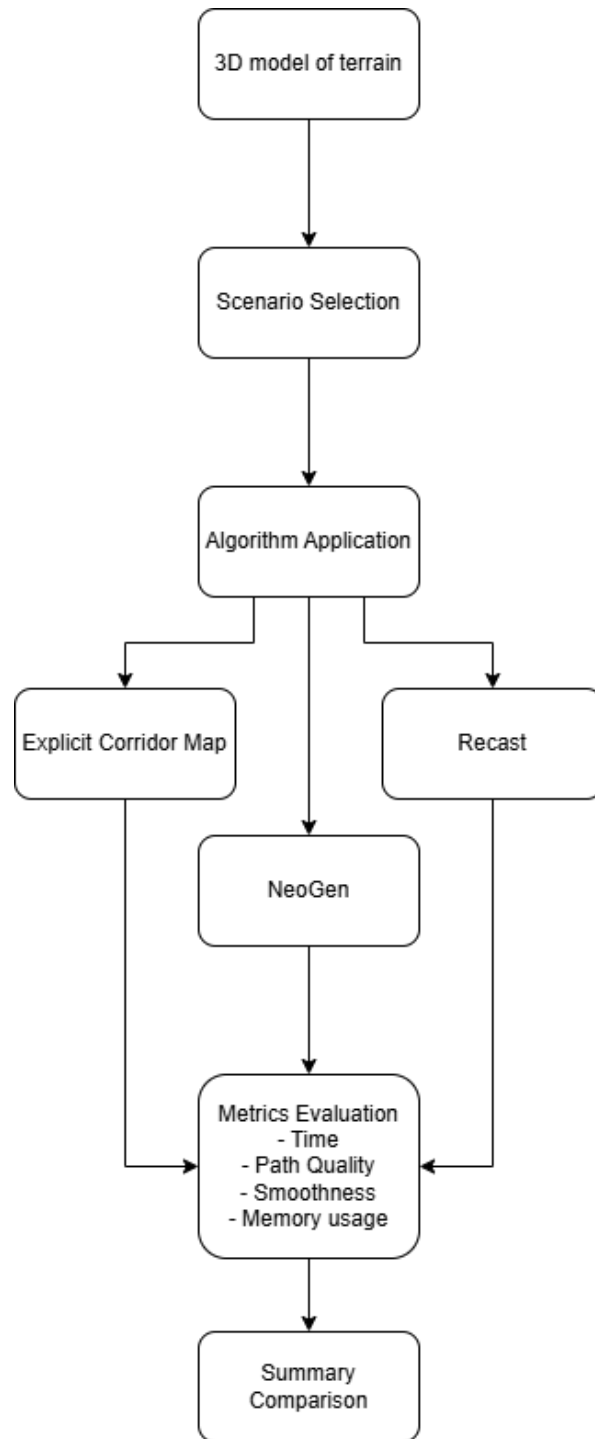


Figure 3: Diagram visualizing methodology.

Category	Parameter	Value	Description
Rasterization	Cell Size	0.05 m	Horizontal size of the cells the terrain is sampled into
	Cell Height	0.02 m	Vertical quantisation of the height field
Agent Parameters	Height	0.303 m	Vertical clearance required by the rover
	Radius	0.2225 m	Half the rover width, used for clearance
	Maximum Step	0.07 m	Largest height difference treated as drivable
	Maximum Slope	20°	Steepest incline treated as drivable
Meshing	Minimum Region Size	30 cells	Smallest connected group of cells kept as a region
	Minimum Obstacle Area	0.198 m ²	Smallest hole kept as an obstacle, equal to the agent footprint (NeoGen)
	Vertices per Polygon	6	Maximum number of vertices per output cell
	Merge Size	20	Threshold for merging adjacent regions (Recast)
	Maximum Edge Length	12	Longest allowed contour edge, in cells (Recast)
	Maximum Edge Error	1.3	Deviation allowed when simplifying contours (Recast)
	Detail Sampling	6 / 1	Height sampling distance and maximum error (Recast)

Table 1: Navigation mesh parameters. Both generators are built from this single set. The entries marked with a tool apply only to that tool, and no parameter differs between the two.

intended for coarse models. On a metric model whose agent is 0.445 m wide that floor is what decides, and it silently discards every boulder smaller than about a square metre. The floor is disabled in the runs reported below, so the agent footprint alone governs. Section 6 quantifies what it costs to leave it in.

5.2 Metrics

The metrics below follow the groups established in the comparative literature [18, 13]. Every quantity is computed on the metric model of Section 4.1, in the ground plane, from the exported meshes rather than from inside either generator, so both tools are measured by the same code.

5.2.1 Reference sets

Coverage is meaningless without a reference, and this study uses two of them, because the two generators do not return the same kind of object. Table 2 collects the symbols, which are used in every result table and in the discussion that follows.

The **drivable surface** F is the union of those triangles of the terrain model whose normal is within the slope limit of vertical, projected onto the ground plane. Of the 87 232 triangles of the model, 73 432 (84.2%) qualify at 20°. Connected pieces and interior holes smaller than the agent footprint are then removed, because both generators filter classification noise at that scale: keeping every two-centimetre steep facet would puncture the reference with pinholes that neither tool – and no rover – treats as an obstacle, and eroding such a reference would erase most of the field. This yields $F = 102.63 \text{ m}^2$.

The **eroded free space** $F \ominus r$ is F shrunk by the agent radius: the set of positions in which the centre of the rover may legally be placed. It measures 66.97 m^2 , so on this field roughly a third of the drivable surface is within one rover radius of something the rover must not touch.

Symbol	Meaning	Value here
r	Agent clearance radius, half the width of the rover	0.2225 m
F	Drivable surface. The ground the rover may drive over: terrain triangles within the slope limit, projected onto the ground plane, with pieces and holes below the agent footprint removed	102.63 m ²
$\ominus$	Erosion (Minkowski difference). $A \ominus r$ is the set of points of A lying at least r away from the boundary of A	operator
$F \ominus r$	Eroded free space, equivalently the configuration space of the rover: the positions in which the centre of the rover may legally be placed	66.97 m ²

Table 2: Notation for the two reference sets. The distinction between them is what the whole comparison turns on: a mesh describing F and a mesh describing $F \ominus r$ are different objects, and each is only interpretable against its own reference.

5.2.2 Metric definitions

- **Coverage Metrics.** Every quantity in this group is deliberately a ratio rather than an area. The absolute extent of a mesh is not comparable between the two tools, because they describe different sets: one covers the ground the rover may drive over, the other the positions its centre may occupy, and the second is necessarily the smaller. Read as fractions of the reference that matches each mesh, the same measurements become comparable.
 - **Normalized Coverage:** the area of the intersection of the navigation mesh with F , divided by the area of F . It answers how much of the drivable surface the mesh found.
 - **Coverage of Eroded Free Space:** the same fraction computed against $F \ominus r$, which is the fair reference for a mesh that already accounts for clearance.
 - **Incorrect Area:** the share of the mesh lying outside F , that is, terrain the mesh claims as drivable but the classification does not support.
 - **Area Outside $F \ominus r$:** the share of the mesh in which the rover’s centre cannot legally stand. For a mesh in configuration space it is near zero by construction. For a mesh describing the drivable surface it is not an error but a measure of the clearance work left to whatever consumes the mesh.
 - **Self-overlap:** the sum of the cell areas minus the area of their union, as a share of that union. A well-formed navigation mesh has none.
- **Convexity Metrics**
 - **Convexity:** cell area divided by the area of its convex hull, averaged over cells. A value of 1.0 means every cell is convex, which both generators guarantee by construction. The metric is reported to confirm the guarantee holds, not to discriminate between them.
- **Adjustment Metrics**
 - **Adjustment:** the share of the mesh area that lies on the drivable surface. It is the precision counterpart of coverage: coverage asks how much of the terrain the mesh found, adjustment asks how much of the mesh is justified by the terrain.
- **Complexity Metrics**
 - **Number of Cells:** convex polygons in the output mesh.
 - **Number of Vertices:** distinct vertex positions, deduplicated to 0.1 mm. Deduplication matters: one exporter writes each cell’s corners separately and the other shares them, so counting raw entries would compare file layout rather than mesh complexity.

- **Number of Portals:** edges shared by exactly two cells, i.e. the arcs of the adjacency graph a planner would traverse.
- **Vertices per Cell:** mean, standard deviation and total, indicating how much detail each cell carries.
- **Cell Area:** mean and median, which together expose whether the mesh is made of comparable pieces or of a few large cells surrounded by slivers.

- **Performance Metrics**

- **Construction Time:** wall-clock time of the build call itself, excluding loading and parsing the terrain model, reported as the median of repeated runs.
- **In-memory Footprint:** bytes needed to hold the mesh – vertex positions, cell rings and adjacency – rather than any proxy derived from its geometry.

- **Compressibility Metrics**

- **Serialized Size:** the mesh in a compact binary encoding (32-bit positions, 32-bit indices), which is what would actually be transmitted.
- **Compressed Size, Compression Ratio, Compression and Decompression Time:** the same payload through zlib at maximum level, timed as the median of twenty runs.

6 Results

Table 4 reports the metrics of Section 5.2 for both generators built from the shared parameter set of Table 1. The Explicit Corridor Map implementation was not completed due to time constraints, thus no results are available for this method.

The single most consequential result is not in any individual row: **the two generators do not return the same object**. NeoGen returns the drivable surface itself. Recast returns the configuration space of the rover, which is that surface already eroded by the agent radius. These are different sets, and their extents are not two measurements of one quantity, so the comparison is stated here in relative terms throughout.

Read against a single reference, one of the two always looks defective. Measured against F , Recast recovers 64.07% of the drivable surface and appears to lose a third of the field, when what it has actually done is refuse to place the rover’s centre within one radius of anything it must not touch. Measured against $F \ominus r$, NeoGen reaches 98.31% and appears to overshoot, when what it is describing is ground rather than legal centre positions. Measured each against the reference that matches its semantics, the two agree closely: NeoGen covers 91.92% of F , Recast 89.27% of $F \ominus r$.

The remaining differences are summarised below.

- **Mesh Complexity:** NeoGen produces 1843 convex cells over 5132 distinct vertices, against Recast’s 186 cells over 364 vertices – an order of magnitude more detail for the same terrain and the same parameters. The cause is visible in the cell areas: NeoGen’s mean cell is 0.054 m^2 but its median is $2.1 \times 10^{-5} \text{ m}^2$. Most of its cells are slivers left along the contours by raster tracing and simplification, which the greedy convex merge cannot absorb without breaking convexity or exceeding the vertex budget: 1314 of the 1843 cells (71.3%) are smaller than 10 cm^2 and together account for 0.06% of the mesh area. Recast simplifies its contours before triangulating and produces none that small, with a median cell of 0.109 m^2 . The distinction matters for a planner that treats one cell as one occupiable place: only 63 of NeoGen’s cells can host the rover at all, and they carry 72.8% of the area, so the mesh needs its cells merged by agent footprint before it can be planned on.
- **Connectivity:** NeoGen exports 1847 portals alongside its cells, so the adjacency graph a multi-agent planner needs comes out of the generator directly, with no additional discretisation of the free space. Recast’s 195 shared edges carry the same information but are recovered from the polygon soup rather than exported as a first-class structure.

- **Convexity:** both meshes are convex cell by cell (1.0), as both algorithms guarantee. The value is worth reporting even though it cannot discriminate, because it confirms that NeoGen’s convex decomposition survives the greedy merge intact and that Recast’s contour simplification never folds a cell.
- **Fidelity:** 0.47% of Recast’s mesh falls outside the drivable surface, for an adjustment of 0.995. The corresponding share of NeoGen’s mesh is 4.99%, for an adjustment of 0.950. The gap is the price of tracing the drivable region at raster resolution and then simplifying its contour, because the simplified boundary cuts corners across terrain the classification rejected. Neither mesh exhibits any self-overlap.
- **Clearance:** 33.70% of NeoGen’s mesh – a third of it – lies outside $F \ominus r$, meaning the rover’s centre cannot legally be placed there even though the ground beneath it is drivable. This is not an error but the semantics of the output. It does mean the mesh cannot be handed to a planner as a set of occupiable places without a clearance step of its own, which Section 6.1 supplies. The corresponding figure for Recast is 9.51%, which is boundary quantisation at the 0.05 m cell size rather than a modelling difference.
- **Performance:** Recast builds the mesh in 0.019 s, NeoGen in 7.87 s, a factor of roughly 420. Part of that is C++ against Python, but the shapes of the two computations also differ: NeoGen runs a constrained triangulation and a pairwise convex merge over thousands of primitives, and the merge dominates.
- **Storage:** the serialized NeoGen mesh is 98.24 KB against Recast’s 7.58 KB, and compresses to 55.75 KB against 3.91 KB. The ratio between the two follows the cell counts rather than anything else, which is what makes it a complexity result and not a storage result. The compression ratios themselves are close (1.76 against 1.94) and both are modest, because a binary encoding of irregular coordinates has little redundancy left to exploit. A high compression ratio on a navigation mesh is in fact a warning sign, as it indicates a grid of repeated identical primitives rather than a decomposition adapted to the terrain.

6.1 Applying the clearance step

The difference in semantics identified above is not a defect to be reported and left standing. It is a missing step, and the step is cheap. Because NeoGen returns the drivable surface, the configuration space of a particular rover is obtained from it by eroding that surface by the agent radius, which is the same operation Recast performs internally before it triangulates. The mesh can therefore be brought into Recast’s frame of reference after the fact, and the comparison repeated where both objects mean the same thing.

Table 3 reports the outcome. Eroding the NeoGen mesh by the 0.2225 m clearance radius takes 0.10 s, roughly one part in eighty of the 7.87 s build, and leaves a region that covers 91.56% of $F \ominus r$ against Recast’s 89.27%. The share of the result in which the rover’s centre cannot legally stand falls from 33.70% to 7.00%, below Recast’s 9.51%, and the share lying outside the drivable surface altogether falls to 0.24%, because eroding a region can only pull its boundary inwards, away from the terrain the classification rejected. In configuration space the two generators therefore agree to about two percentage points on this field, and the residual difference favours the eroded NeoGen mesh. The first row of the table makes the same point from the other direction. Once the clearance step has been applied, the NeoGen mesh scores 64.09% against F and Recast 64.07%, so the figure that looked like a decisive weakness of Recast turns out to be a property of any mesh in configuration space measured against the wrong reference.

Two caveats keep this from being a free win. The erosion returns a region, not a convex decomposition, so a mesh usable by a planner has to be re-triangulated and re-merged from that region, and the cell counts and portal counts of Table 4 do not carry over. The erosion also splits the field into six connected components where the unclamped surface was one, which is the correct answer rather than a loss: a corridor narrower than the rover is not a corridor, and a surface-space mesh silently reports it as passable.

The practical consequence is that the agent-agnostic output remains the more general starting point. One drivable surface serves a heterogeneous fleet, and each rover derives its own configuration space from it for a tenth of a second of work, instead of the terrain being voxelized again per vehicle.

Metric (%)	NeoGen	NeoGen + clearance	Recast
Coverage of F	91.92	64.09	64.07
Coverage of $F \ominus r$	98.31	91.56	89.27
Outside F (of mesh)	4.99	0.24	0.47
Outside $F \ominus r$ (of mesh)	33.70	7.00	9.51

Table 3: The NeoGen mesh before and after an explicit clearance step. The eroded mesh and Recast both describe the positions the rover’s centre may occupy, so their columns are directly comparable and the first row, measured against the wrong reference for both of them, is equally misleading about both. Table 2 defines F and $F \ominus r$.

6.2 Effect of the minimum obstacle area

The threshold below which a hole is discarded as noise rather than kept as an obstacle (Section 5.1) is worth isolating, because it moves the metrics in opposite directions. Rebuilding with the absolute 1 m² floor left in place, only 5 of the 14 obstacles survive: the boulders and the smaller ripple structures are absorbed into the drivable region. Coverage of F rises from 91.92% to 92.99%, which in isolation reads as an improvement. At the same time the incorrect area rises from 4.99% to 6.13% of the mesh, the share outside $F \ominus r$ from 33.70% to 34.70%, and adjustment falls from 0.950 to 0.939. The mesh did not get better, it got larger by claiming obstacles as floor. It is a compact illustration of why a coverage figure cannot be read on its own.

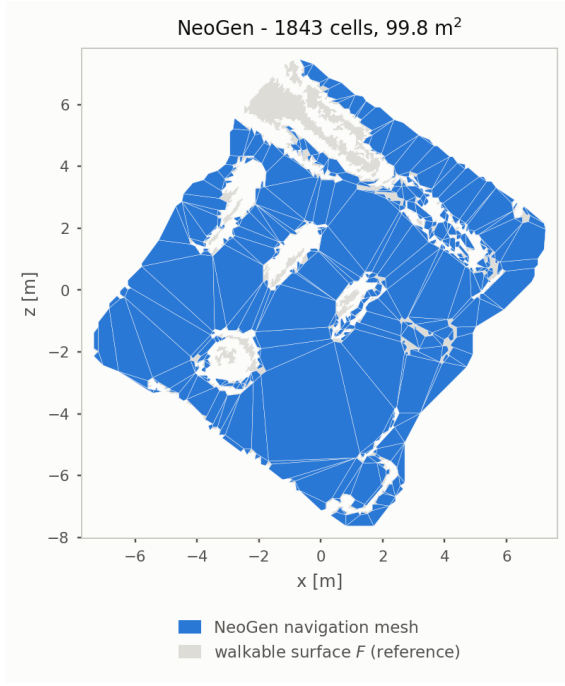


Figure 4: Navigation mesh generated by NeoGen (blue) over the drivable surface F (grey). The mesh follows the drivable region up to its boundary and around the obstacles retained after noise filtering. The grey areas that remain are the ripple structures and crater rims the classification rejected.

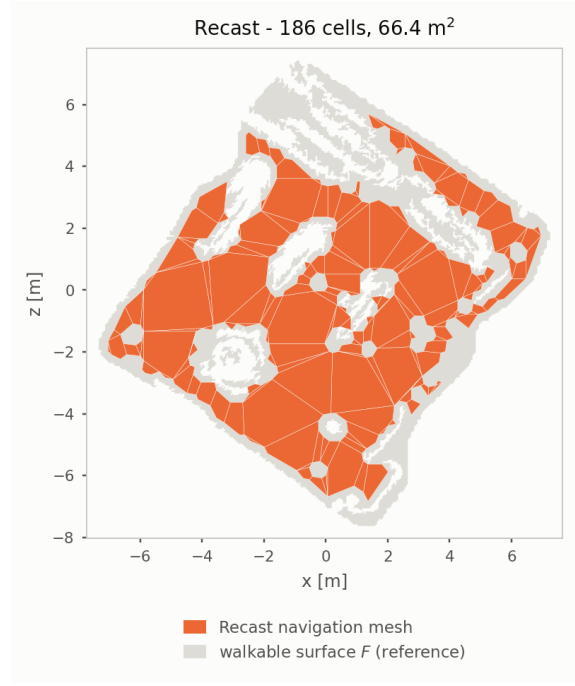


Figure 5: Navigation mesh generated by Recast (orange) over the same reference. The uniform grey margin along the field boundary and around every obstacle is the agent-radius erosion: Recast returns the positions the rover’s centre may occupy, not the ground it may drive over.

Metric	Units	NeoGen	Recast
Number of Cells	-	1,843	186
Number of Vertices	-	5,132	364
Number of Portals	-	1,847	195
Obstacles Retained	-	14	n/a
Vertices per Cell (mean)	-	4.79	4.05
Vertices per Cell (std. dev.)	-	1.23	1.10
Total Cell Complexity	-	8,831	754
Convexity	-	1.00	1.00
Mean Cell Area	m ²	0.054	0.355
Median Cell Area	m ²	0.00002	0.109
Self-overlap (of mesh)	%	0.00	0.00
Normalized Coverage (against F)	%	91.92	64.07
Coverage of $F \ominus r$	%	98.31	89.27
Incorrect Area (of mesh)	%	4.99	0.47
Area Outside $F \ominus r$ (of mesh)	%	33.70	9.51
Adjustment	-	0.950	0.995
Construction Time	s	7.87	0.019
In-memory Footprint	KB	112.67	9.10
Serialized Size	KB	98.24	7.58
Compressed Size	KB	55.75	3.91
Compression Ratio	-	1.76	1.94
Compression Time	ms	5.24	0.32
Decompression Time	ms	0.31	0.03

Table 4: Metrics of the two navigation meshes, both built from the shared parameters of Table 1 on the same metric terrain model. The coverage and fidelity rows are given as fractions rather than as areas, because the two meshes describe different sets and their extents are not two readings of one quantity. Table 2 defines F and $F \ominus r$ and gives the area of each, so any of these fractions can be converted back to square metres.

7 Discussion

The implementation and testing of navigation meshes for Martian terrain revealed several key challenges and considerations:

Resolution Trade-offs: A critical challenge lies in selecting the appropriate resolution for terrain model generation. Higher resolution models provide more detailed terrain representation but can lead to issues with normal calculations and mesh generation. Conversely, lower resolution models process more efficiently but risk missing critical obstacles or terrain features. A promising solution to this trade-off could be a hybrid approach combining low-resolution base terrain with high-resolution obstacle data. Using methods like RANSAC (Random Sample Consensus) [5], key obstacles and terrain features could be identified from high-resolution scans and integrated into a lower-resolution base map. This approach would maintain computational efficiency while preserving critical navigation information.

Mesh Generation Limitations: Navigation meshes have one fundamental issue – often, in order to reach a specific location in a particular way, one must pass through an area that may be marked as impassable in the navigation mesh. Additionally, validation and human intervention are required to correct the edges of the created mesh and fix errors within it.

Surface Space versus Configuration Space: The most practically important finding of this comparison is that two tools presented in the literature as alternatives for the same job return different objects. Recast bakes the agent radius into the mesh, so any point of the output is a legal position for the rover’s centre and a path through the mesh needs no further clearance reasoning, at the price of a mesh that must be rebuilt whenever the agent changes. NeoGen returns the drivable surface itself, which is agent-agnostic and reusable across a heterogeneous fleet, but a third of that surface consists of positions where a 0.445 m rover cannot legally stand.

Neither convention is wrong, and the gap between them is closed by an operation rather than by a choice. Eroding the surface-space mesh by the agent radius, as Section 6.1 shows, costs a tenth of a second and lands within two percentage points of Recast in Recast’s own frame of reference. What remains wrong is doing none of this: comparing meshes built under different conventions against a single reference, or feeding a surface-space mesh to a planner that assumes every cell is a place an agent can occupy. The second error is the dangerous one on real terrain, because it does not announce itself. The planner returns a path through a gap the rover cannot physically enter, and nothing in the mesh says otherwise.

For the heterogeneous fleet this work targets, the surface-space output is therefore the better starting point, on the condition that the clearance step is an explicit stage of the pipeline with a named agent attached to it, rather than an assumption inherited from whichever tool produced the mesh.

Obstacle Filtering Thresholds: Noise filtering on the classification is unavoidable on a real scan, but its threshold is a modelling decision with a physical meaning, not a numerical convenience. An absolute floor of one square metre, harmless on a coarse city model, erases every boulder on a 10×10 m field with a half-metre rover, and does so while improving the coverage figure. Thresholds of this kind should be expressed in units of the agent, and any coverage number should be reported alongside a precision counterpart that a spurious enlargement of the mesh would degrade.

Implementation Considerations: Recast builds the mesh some two orders of magnitude faster than the NeoGen implementation used here. Part of that gap is C++ against Python and would narrow with a compiled implementation, but not all of it: the constrained triangulation and the pairwise convex merge over several thousand primitives are inherently more work than Recast’s contour simplification, and the merge dominates the runtime. Neither figure is a problem at the scale of this field, where a full rebuild takes seconds, but both would matter for on-board incremental updates.

Scalability Challenges: The height field on which the classification runs is a natural fit for GPU execution, and it is already computed there in the implementation used here. The stages that dominate the runtime, triangulation and convex merging, are the sequential ones. Scaling to full Martian terrain therefore depends less on raw parallelism than on making the merge incremental, so that a local terrain update revisits only the cells it touches instead of rebuilding the mesh.

User Interface and Validation Requirements: The generated navigation meshes require human validation, particularly in critical areas where precise navigation is essential, similar to current Mars rover operations where human operators verify and adjust planned operations before execution [17]. The collaborative nature of Martian exploration, where mission objectives and navigation points

are determined by scientists [7], necessitates an integrated, user-friendly graphical interface. Such an interface should combine navigation mesh visualization with mission planning capabilities and validation tools, allowing scientists to designate exploration targets, review generated paths, and verify navigation mesh quality. This would create a streamlined workflow between scientific planning and navigation implementation.

8 Future Work

This study establishes a foundation for evaluating navigation meshes in Martian analog environments. Future research directions include the implementation of more advanced scenarios, with a set of test cases designed to represent narrow corridors, open spaces, and regions with varying slopes. Another avenue involves dynamic terrain adaptation, where real-time updates to the navigation mesh reflect changing terrain conditions. Energy efficiency metrics will also be introduced to measure rover energy consumption during navigation. Additionally, multi-agent navigation will be explored, extending evaluation metrics to assess coordination and collision avoidance among multiple rovers.

A comprehensive series of A* pathfinding tests will simulate realistic Martian challenges to evaluate shortest path quality (SPQ) metrics. These scenarios will involve navigating narrow passages, ascending steep slopes, traveling across flat terrain, and traversing rocky environments with multiple obstacles. Such tests aim to validate the robustness and practical applicability of navigation meshes across diverse operational contexts.

9 Conclusion

This study integrates comparative metrics from [18] to evaluate the suitability of navigation meshes for Martian exploration, on a reconstruction of a Martian analog field and with the parameters of a real rover platform. Built from one shared parameter set on one metric terrain model, both generators reproduce the drivable terrain closely. NeoGen covers 91.92% of the drivable surface, Recast 89.27% of what remains of that surface once it is eroded by the rover radius. The two figures are not in competition, because they are measured against different references and describe different objects, and the comparison only becomes meaningful once that is said out loud. Applying an explicit clearance step to the NeoGen mesh brings it into Recast’s frame of reference for a tenth of a second of work, after which the two agree to within about two percentage points.

For a mission planner the practical trade-off is therefore not accuracy but what the mesh is for. Recast is the better choice when a single agent geometry is fixed in advance: it is some 400 times faster to build, an order of magnitude smaller to store and transmit, places almost nothing outside the drivable terrain (0.47% of its area, adjustment 0.995), and needs no clearance reasoning downstream. NeoGen is the better choice when the same terrain must serve a heterogeneous fleet, or when the planner wants the topology rather than the polygons: it returns the drivable surface independently of any particular rover, together with 1847 explicit portals over its 1843 convex cells, which is a planning graph rather than a polygon soup. Its costs are equally concrete: 4.99% of its area is not supported by the classification, a third of it is not legally occupiable by the rover’s centre, and most of its cells are contour slivers holding a negligible share of the area. Two post-processing stages therefore stand between that output and a planner. The clearance step is measured here and is cheap, while merging the slivers by agent footprint remains to be done.

The comparison also carries a methodological lesson. Coverage read on its own rewards a mesh for growing, whichever direction it grows in: loosening the obstacle filter here raised coverage by a point while making the mesh measurably less correct on every other axis. A coverage figure is interpretable only next to a precision counterpart, against a reference whose semantics matches the mesh, and computed by a measurement path that has itself been checked.

The successful reconstruction and testing of navigation meshes in the Józefosław research field illustrates the practical challenges and preprocessing required for generating accurate navigation meshes – including the mundane one, unit scale, which quietly invalidates every area figure and every agent dimension when it goes unstated. Future research will involve refining metrics, optimizing implementations, and conducting more extensive testing in real-world Martian analogs.

References

- [1] Ronald C Arkin. “Navigational path planning for a vision-based mobile robot”. In: *Robotica* 7.1 (1989), pp. 49–63.
- [2] Max Bajracharya, Mark W. Maimone, and Daniel Helmick. “Autonomy for Mars Rovers: Past, Present, and Future”. In: *Computer* 41.12 (2008), pp. 44–50. DOI: 10.1109/MC.2008.479.
- [3] Steven C Bankes. “Agent-based modeling: A revolution?” In: *Proceedings of the National Academy of Sciences* 99.suppl.3 (2002), pp. 7199–7200.
- [4] Jeffrey J. Biesiadecki et al. “Mars Exploration Rover Surface Operations: Driving Opportunity at Meridiani Planum”. In: *IEEE Robotics & Automation Magazine* 14.1 (2007), pp. 63–71. DOI: 10.1109/MRA.2007.339616.
- [5] Martin A Fischler and Robert C Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.
- [6] J. P. Grotzinger et al. “Mars Science Laboratory Mission and Science Investigation”. In: *Space Science Reviews* 170.1-4 (2012), pp. 5–56. DOI: 10.1007/s11214-012-9892-2.
- [7] John P Grotzinger et al. “Curiosity’s mission of exploration at Gale Crater, Mars”. In: *Elements* 11.1 (2015), pp. 19–26.
- [8] Marcelo Kallmann and Mubbasir Kapadia. “Navigation meshes and real-time dynamic planning for virtual worlds”. In: *ACM SIGGRAPH 2014 Courses*. SIGGRAPH ’14. Vancouver, Canada: Association for Computing Machinery, 2014. ISBN: 9781450329620. DOI: 10.1145/2614028.2615399. URL: <https://doi.org/10.1145/2614028.2615399>.
- [9] *Leo Rover Documentation: Technical Specification, rev. 1.9*. 2024. URL: <https://docs.fictionlab.pl/leo-rover/documentation/specification> (visited on 09/10/2026).
- [10] Mark Maimone, Yang Cheng, and Larry Matthies. “Two Years of Visual Odometry on the Mars Exploration Rovers”. In: *Journal of Field Robotics*. Vol. 24. 3. 2007, pp. 169–186. DOI: 10.1002/rob.20184.
- [11] Larry Matthies et al. “Computer Vision on Mars”. In: *International Journal of Computer Vision* 75.1 (2007), pp. 67–92. DOI: 10.1007/s11263-007-0046-z.
- [12] M Mononen. “Github Recast and Detour”. In: *Navigation-mesh toolset for games, viewed February* (2025).
- [13] Ramon Oliva and Nuria Pelechano. “NEOGEN: Near optimal generator of navigation meshes for 3D multi-layered environments”. In: *Computers & Graphics* 37.5 (2013), pp. 403–412.
- [14] Science in Poland. *Researchers build testing field for Mars rovers in Józefosław near Warsaw*. 2024. URL: <https://scienceinpoland.pl/en/news/news%2C103754%2Cresearchers-build-testing-field-mars-rovers-jozefoslaw-near-warsaw.html> (visited on 01/20/2024).
- [15] Benjamin Rothrock et al. “SPOC: Deep Learning-based Terrain Classification for Mars Rover Missions”. In: *AIAA SPACE Forum*. 2016. DOI: 10.2514/6.2016-5539.
- [16] Wouter van Toll et al. “The explicit corridor map: Using the medial axis for real-time path planning and crowd simulation”. In: *32nd International Symposium on Computational Geometry (SoCG 2016)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik. 2016.
- [17] Julie Townsend, Douglas Klein, and Sawyer Brooks. “Robotics verification and validation strategies for perseverance rover sampling and caching”. In: *2022 IEEE Aerospace Conference (AERO)*. IEEE. 2022, pp. 1–13.
- [18] Wouter Van Toll et al. “A comparative study of navigation meshes”. In: *Proceedings of the 9th International Conference on Motion in Games*. 2016, pp. 91–100.
- [19] Wouter G Van Toll, Atlas F Cook IV, and Roland Geraerts. “A navigation mesh for dynamic environments”. In: *Computer Animation and Virtual Worlds* 23.6 (2012), pp. 535–546.
- [20] Robert Wack and Anna Wimmer. “Digital terrain models from airborne laserscanner data-a grid based approach”. In: *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences* 34.3/B (2002), pp. 293–296.