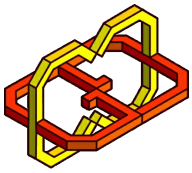


LUNAR INFRASTRUCTURE FOR SIMULATIONS OF LANDING SCENARIOS PROJECT

Yuli Nagel
Kostya Kishnevsky

Supervised by:
Michal Edelstein
Boaz Sternfeld

Spring 2023

Table of contents:

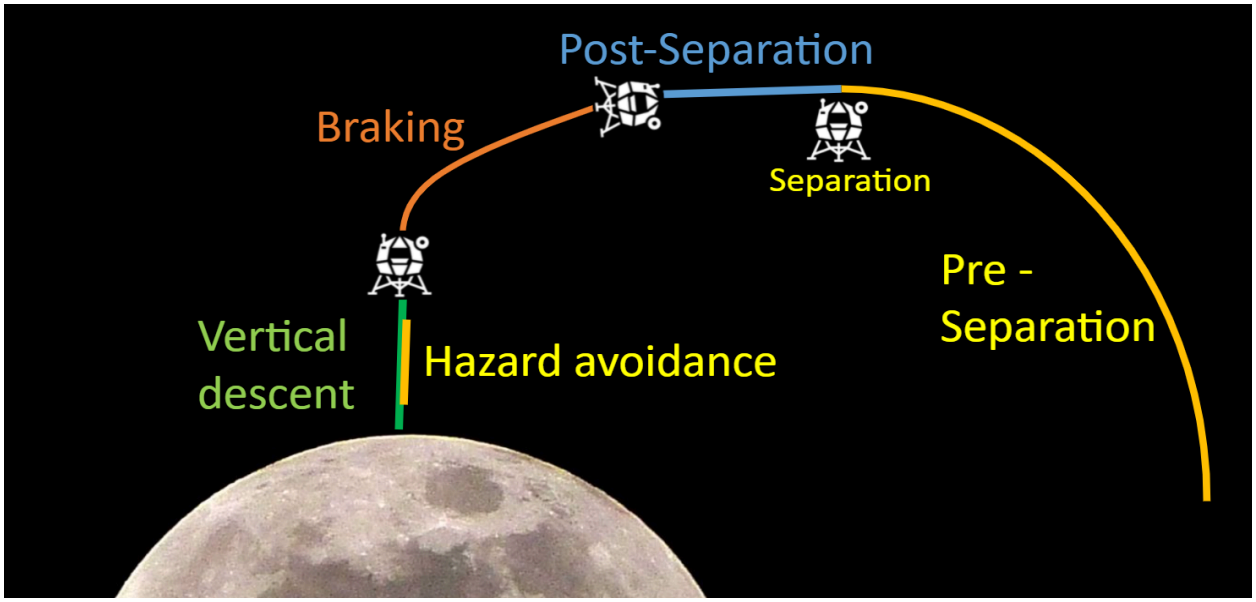
Introduction:

In 2019 SpaceIL launched the first Israeli spacecraft – Beresheet 1. Beresheet 1 is Israel's first-ever spaceship made by SpaceIL in collaboration with Israel Aerospace Industries (IAI). It was sent into space on February 22, 2019, and about a month and a half later, on April 4th, it successfully got into an elliptical lunar orbit around the moon. About a week after getting into orbit, there were some problems during the landing that led to a hard landing at a speed of 1 km. per second onto the moon's surface.

Soon after, the development of Beresheet 2 was announced, and Lulav Space was selected to aid in developing an advanced navigation solution. The new mission was scheduled for 2025 with an aim to conduct a double landing on the Moon and continue in orbit for five years as a platform for education and science activities.

Lulav Space is an Israeli robotics company specializing in space applications. This project enables the generation of 3D datasets and simulations from NASA's real Digital Terrain Models (DTMs), all of which will help to assess Lulav's computer vision navigation algorithms.

Lulav's system comprises 3 space-qualified narrow FOV cameras, and one space-qualified high-rate image processing unit, executing computer-vision-based navigation algorithms.

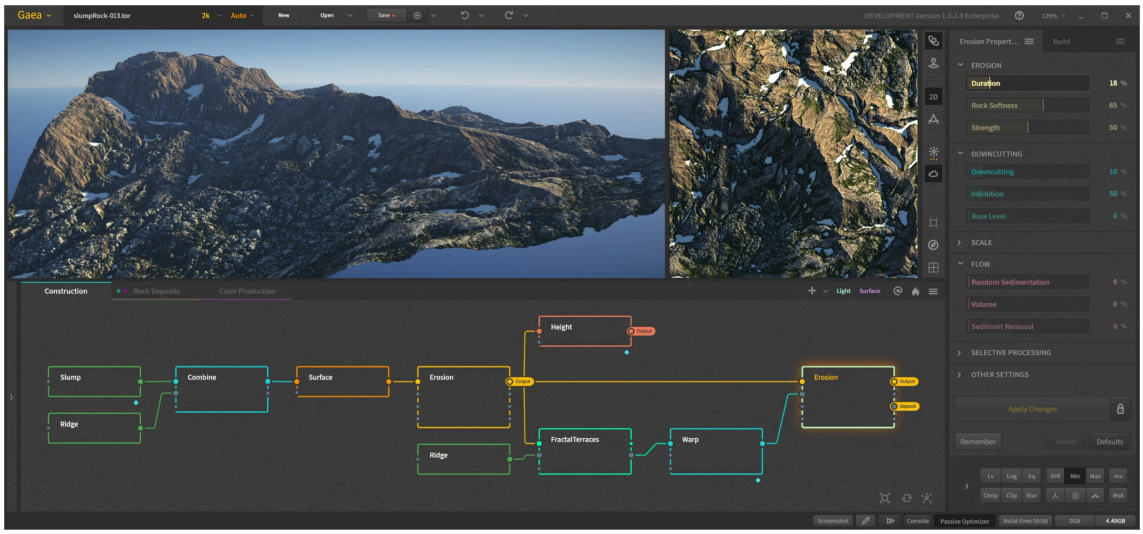


During the Braking phase, the LULAV system will receive pictures from a wide FOV camera pointing to the Moon's surface.
This project focus is to create a dataset to assess Lulav's navigation solution for the Braking stage.

Technologies and Platforms:

- **NASA Lunar Topography SLDEM2015 Dataset:**
  NASA's Lunar Orbiter Laser Altimeter aboard the Lunar Reconnaissance Orbiter (LRO) has collected 43,200 TC DTM tiles to produce a combined topographic map of the Moon at a resolution of 512 ppd. NASA's digital terrain model (DTM) dataset is used to produce mesh tiles dataset for the simulation. The DTMs are segmented into smaller tiles, which serve as the basis for generating high-resolution meshes. This segmentation later enables the execution of a real-time dynamic simulation wherein dynamic imports are employed based on the input scenario - position of the starship in time.

- **GAEA**:
  GAEA is a terrain design tool that enables the creation of a 3D obj mesh from an input TIFF file. The addition of synthetic data in GAEA enables synthetic increase in the terrain resolution with adding details to the surface such as small craters and rocky soil texture.



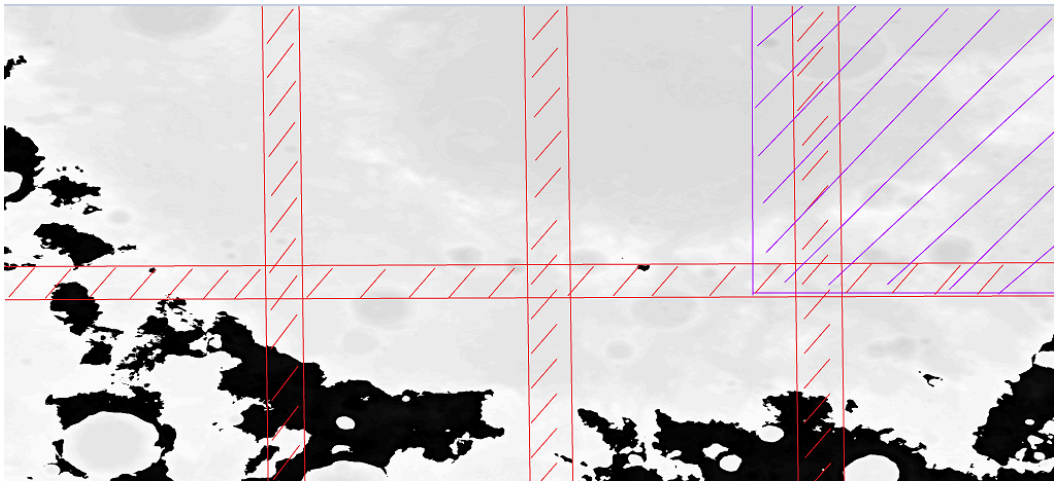
- **Python**:
  Python has many useful scientific libraries such as numpy, pymesh, gdal, which were used for preprocessing and generating the dataset.

- **Blender Rendering**:
  Blender is a 3D computer graphics software tool that enables the creation of the simulation via python API. Using camera locations that are given as an input, Blender is used for the creation of an animation of the camera's movement over time. This process involves rendering each frame, capturing details such as light, shadow, and texture.

Dataset Generation Overview:

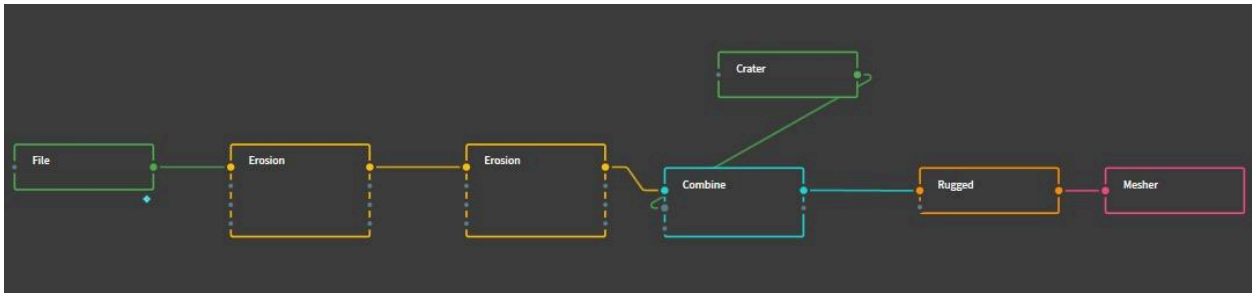
Generating a tiled dataset:

The SLDEM2015 dataset from NASA contains global Digital Elevation Models (DEMs) in FLOAT_IMG format, covering +/- 60 degrees and with a resolution of 512 pixels per degree. Using the GDAL Python library, these dataset tiles are transformed into TIFF files. Subsequently, the TIFF tiles are further divided into smaller tiles utilizing the Rasterio Python library. These smaller tiles are configured to have overlapping margins, enabling a seamless tiling process during the pre-processing phase.
Tiles are saved with a formatted name that contains the tile resolution in pixels, width, height and top left coordinates.



DTM to mesh:

The generated TIFF tiles are processed with GAEA. The TIFF is imported as the base node of the terrain and synthetic details are applied with GAEA filters. This enables adding texture and details to the terrain such as erosion, ruggedness and additional craters.
The terrain is exported as an OBJ file. The exported terrain retains consistent parameters, aligning with the original size and height of the terrain in meters. The output mesh maintains the same ratio of vertices to meters as the entire dataset, ensuring coherence in the pre-process phase.



*Gaea's node graph

Mesh preprocessing:
The meshes are preprocessed to create seamless tiles. Using Pymesh Python library, the OBJ files, (generated using GAEA), are imported as NumPy vertices and faces arrays.
The overlapping areas of the tiles are calculated, and new vertices in these areas are calculated, using a weighted average between the two tiles. This adjustment ensures that vertices closer to one tile have a larger influence, so that the new vertices will fit seamlessly in the original mesh.

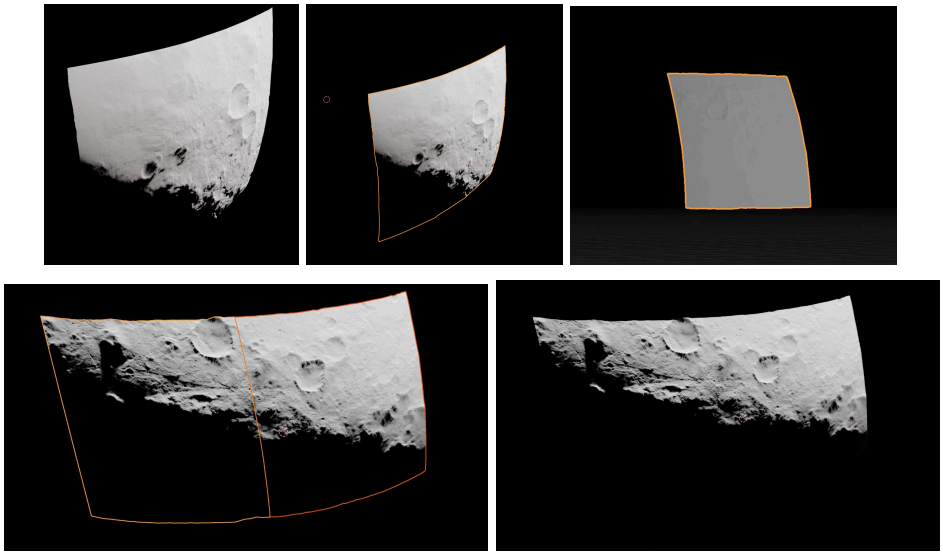
Mesh Transformation:
The meshes are composed of vertices that are based on latitudinal coordinates. Using NASA's meta-data of the appropriate mesh and SpiceyPy Python latrec function, each vertex is converted to rectangular coordinates;

$$latitude = \frac{vertex.x\ [km]}{map\_scale[\frac{km}{pix}]*pix\_deg[\frac{pix}{deg}]} + MINIMUM\_LATITUDE$$

$$longitude = \frac{vertex.y\ [km]}{map\_scale[\frac{km}{pix}]*pix\_deg[\frac{pix}{deg}]} + WESTERNMOST\_LONGITUDE$$

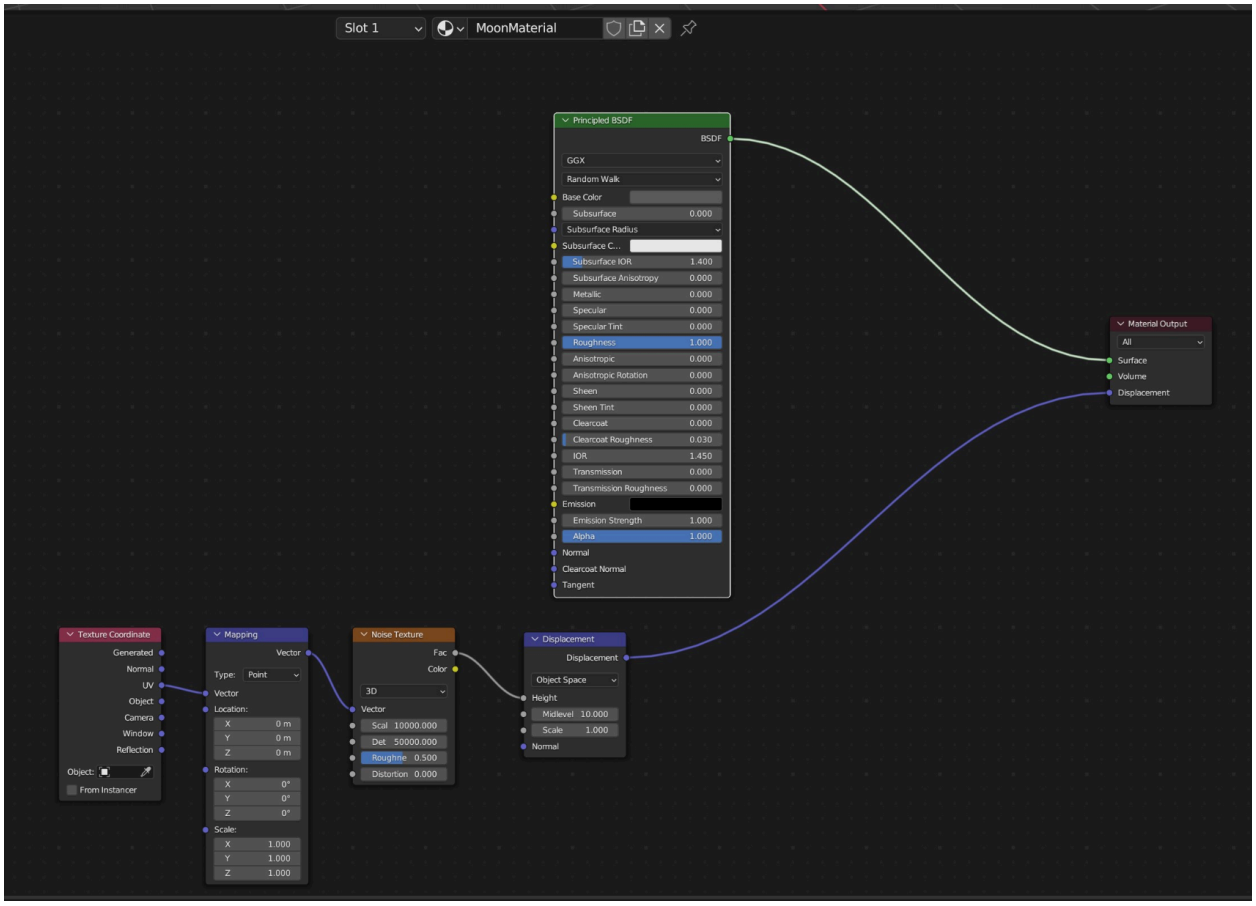
vertex.x - vertex x value in [km]
vertex.y - vertex y value in [km]
map_scale - map scale, taken from the tile's metadata (0.0592252938 [km/pix])
pix_deg - pixel per degree, taken from the tile's metadata (512 [pix/deg])
MINIMUM_LATITUDE - tile's minimum latitude, taken from the tile's metadata [deg]
WESTERNMOST_LONGITUDE - tile's westernmost longitude, taken from the tile's metadata [deg]

A new mesh is constructed, and the surface is now a part sphere with accurate representation of the moon.



Dataset descriptor:
Updating the dataset descriptor is required so that the newly generated objects can be recognizable by the pipeline. The obj names must follow a format to preserve the meta-data of each tile.

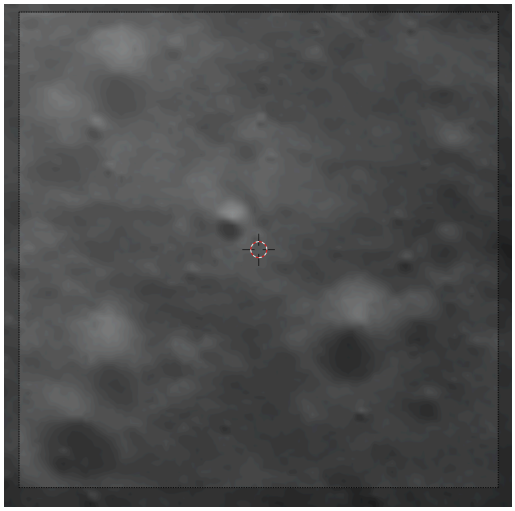
Render Simulation
In the final step, the simulation is rendered with Blender Python API.
An input scenario in JSON format guides the generation and rendering of Blender frames. Each frame is associated with a ground trace from the input camera, and the corresponding meshes are imported. Dynamic textures and smoothing are applied to these imported meshes that enable more details and higher resolution in the rendered image. These textures are based on the moon's true Hapke parameters - a set of parameters that is used to describe the directional reflectance properties of the moon.
To optimize memory usage, any meshes stored in Blender but irrelevant to the current frame are systematically removed, ensuring a streamlined and efficient scene.
The simulation is very versatile, as all parameters, including camera exposure, field of view (FOV), render resolution, and numerous other aspects, can be dynamically controlled and adjusted using the scenario JSON. This level of flexibility ensures a highly customizable, high-detailed and true to life simulation.

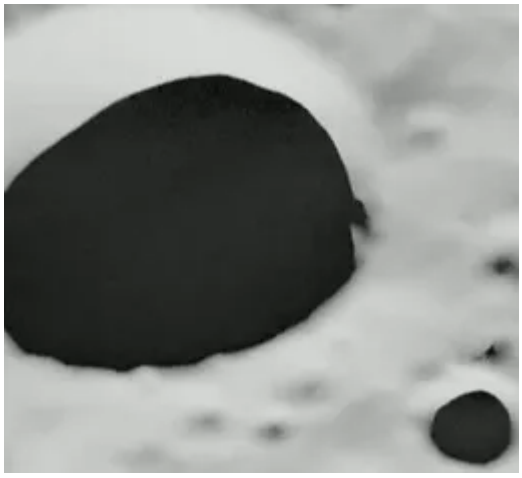


*Blender's shading graph

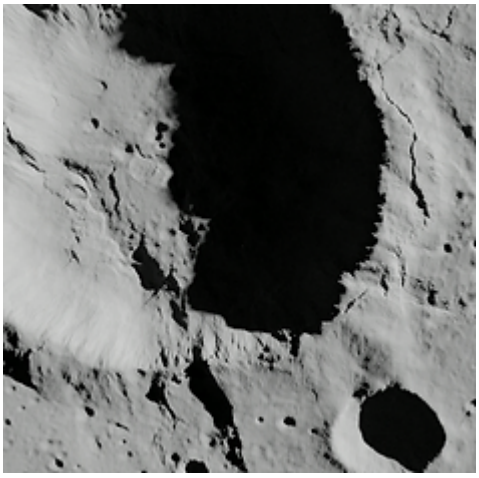
Challenges:

Finding a solution for low resolution DEMs:
NASA's DTMs dataset does not have sufficient resolution for assessing Lulav's computer vision navigation algorithms.
To overcome this challenge, we decided to add synthetic data. Synthetic details are added with softwares such as Gaea and Blender for achieving synthetic higher resolution. Small craters, rocks and different kinds of realistic terrain textures are applied as explained above.





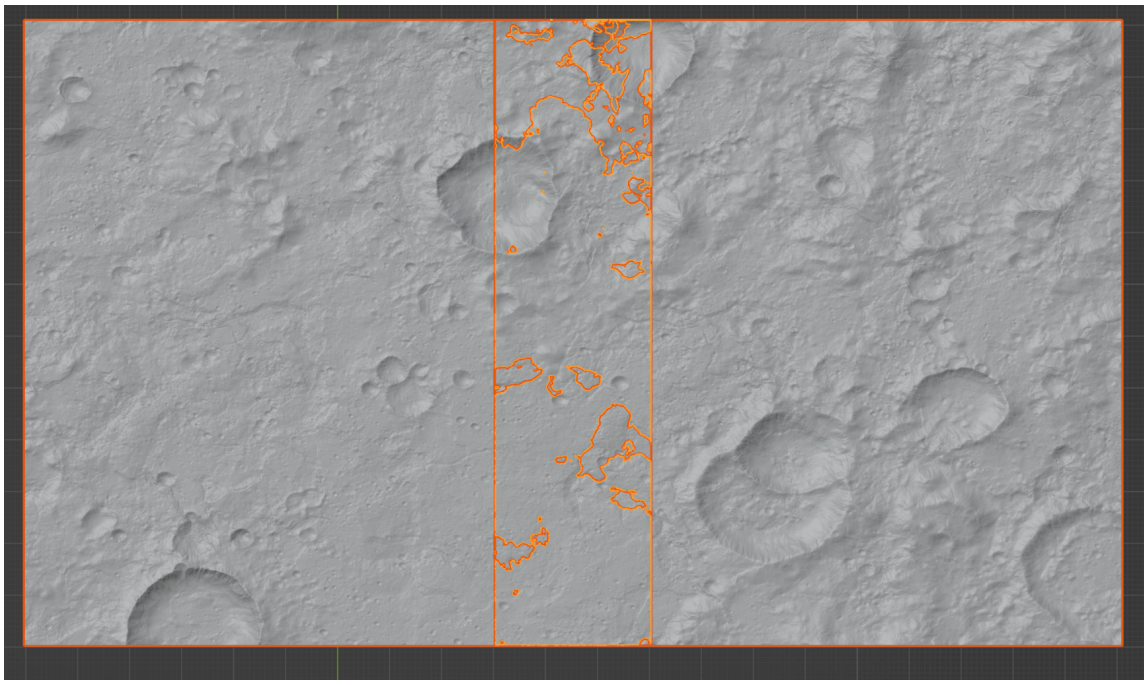
Before



After

Memory limitations:
NASA's DTMs are initially divided into tiles spanning approximately 909x909 kilometers. However, the simulation demands a highly detailed surface, making the generation of meshes from these large tiles memory-intensive and unsuitable for the simulation's requirements. To solve this challenge, the original tiles are cropped into smaller segments, allowing for the creation of more manageable meshes. These smaller meshes are dynamically imported into the simulation and removed as necessary to align with the camera's ground trace, ensuring an optimized representation of the terrain in every frame.
The imported mesh is selected using the dataset descriptor, which contains all the necessary coordinate data. This selection process ensures that the mesh or meshes aligning with the camera's ground trace are specifically imported into Blender using the Python API, allowing seamless integration into the simulation.

Seamless tiling:
Braking simulations require large-scaled and highly detailed terrains that are seamlessly tiled. However, this cannot be done because of the synthetic data addition that creates a visible seam when the tiles are put back together for the simulation. Needless to say, such seams disrupt the assessment of the navigational algorithm since the simulated surface is not consistent with the real surface of the moon.
To overcome this challenge, preprocessing is executed using python. The meshes are loaded as numpy arrays with Pymesh library. Scaling and transformations are applied, and a weighted average is computed between overlapping meshes to create seamless tiles.



Future projects proposals:

- **Multi-resolution dataset**
  Creation of a dataset that involves meshes with different distribution of vertices; finding a solution to weighted average for different ratio of vertices spread. Implementing dynamic import according to camera's height and the resolution of the mesh.
  It's possible to use NASA's higher-resolution NAC DTM's dataset for the smaller high-resolution meshes which are narrow and not sufficient for a whole simulation.

- **ML for resolution increase**
  Creation of textures and resolution increment in the terrains details by using machine learning that trains on NASA's higher-resolution NAC DTM's dataset and applies it on the WAC DTM dataset.

References:

https://www.eng.spaceil.com/beresheet-1
https://www.eng.spaceil.com/post/ramon-space-and-lulav-space-partner-to-navigate-spaceil-s-next-2-lunar-landings
https://lulav.space/
https://pgda.gsfc.nasa.gov/products/54
http://imbrium.mit.edu/DATA/SLDEM2015/
Barker, M. K., Mazarico, E., Neumann, G. A., Zuber, M. T., Haruyama, J., Smith, D. E. "A new lunar digital elevation model from the Lunar Orbiter Laser Altimeter and SELENE Terrain Camera," Icarus, Volume 273, p. 346-355. http://dx.doi.org/10.1016/j.icarus.2015.07.039