Terrain Renderer for Sensor Simulations
-An Accuracy Analysis-

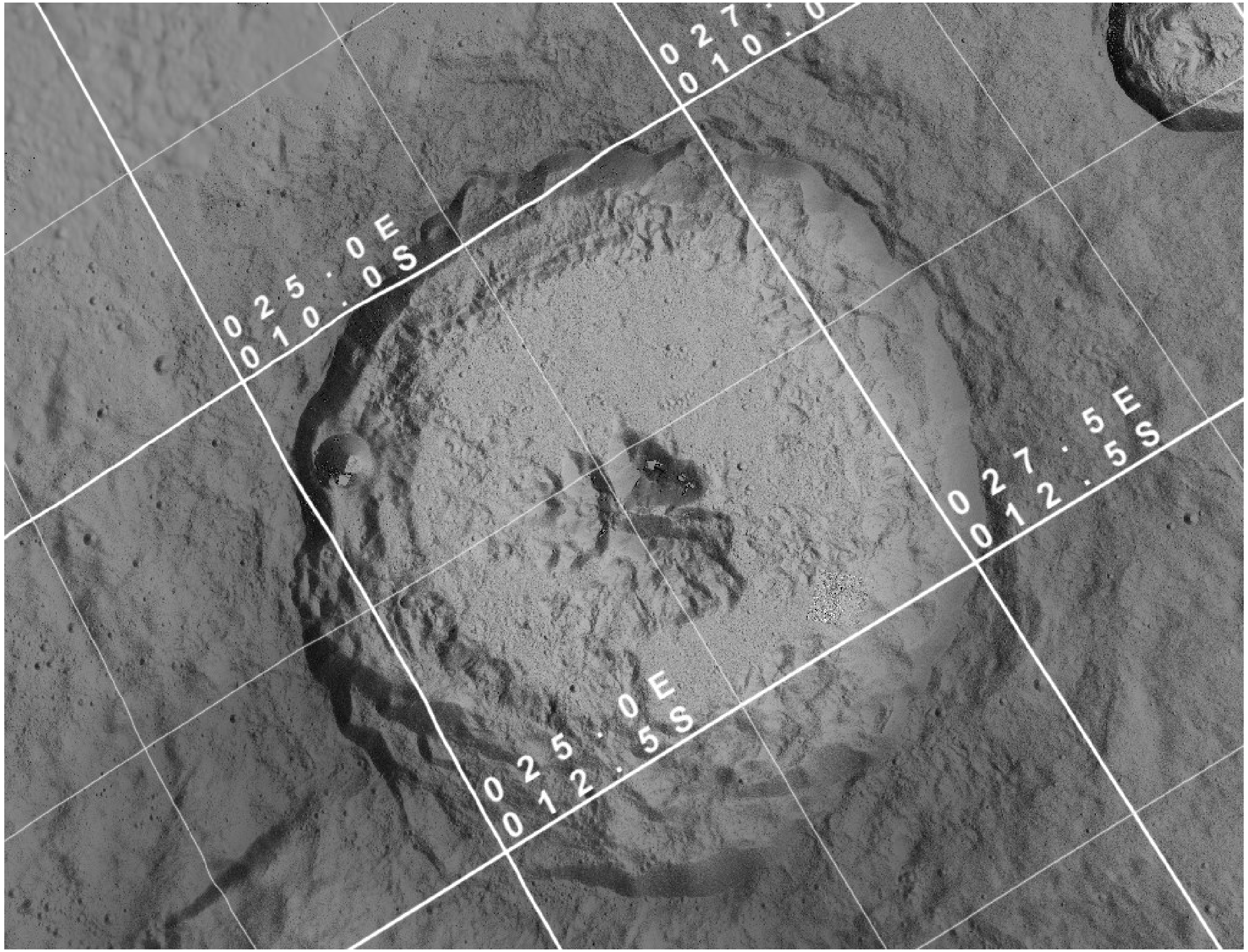
MEON Workshop 2014

Turgay Aslandere

DLR

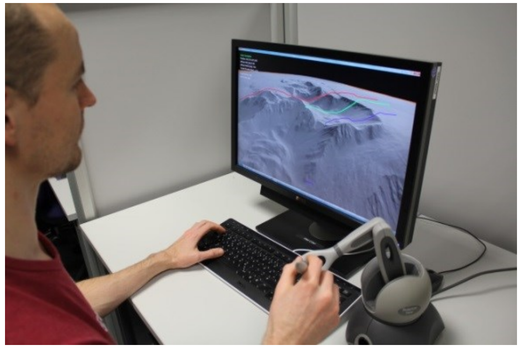
Outline

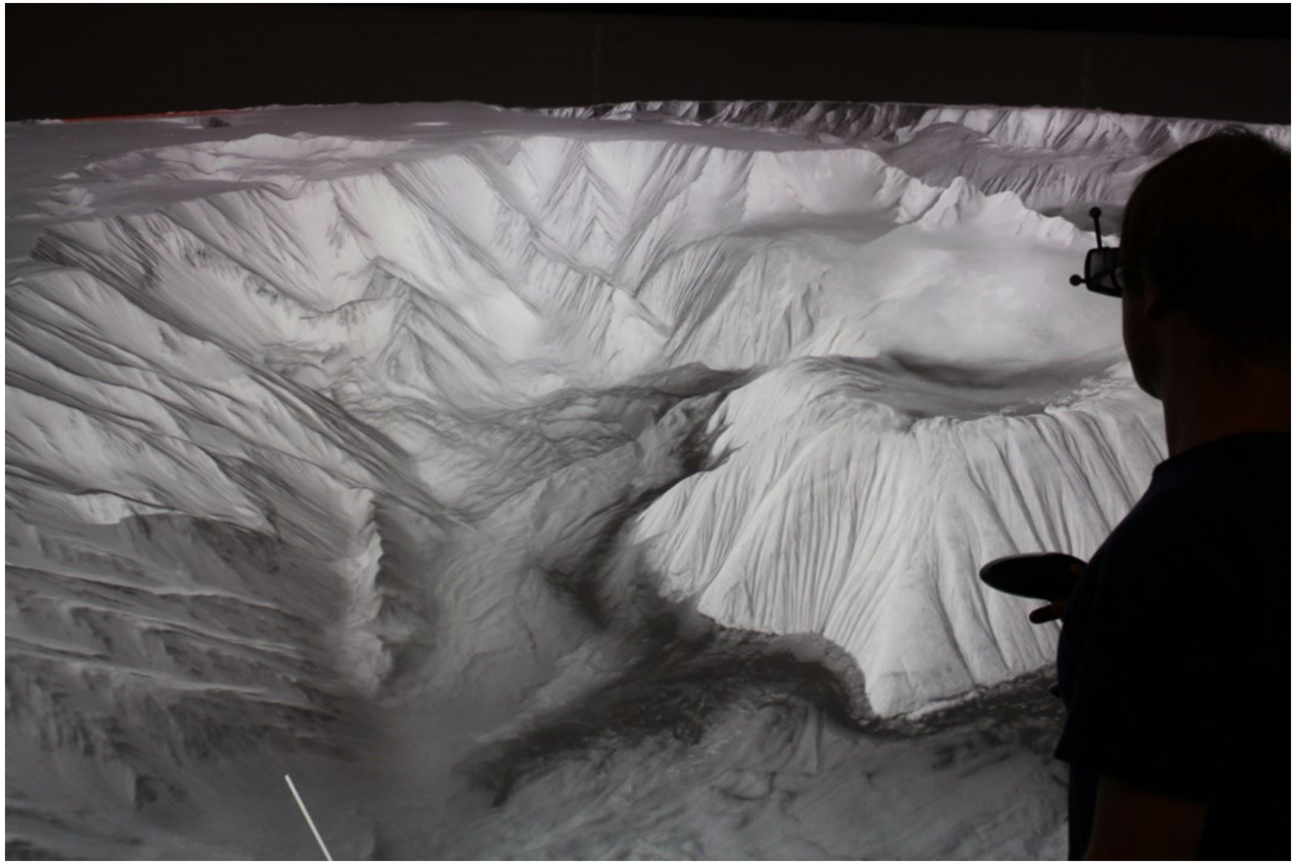
- Application – Terrain Renderer
- Motivation for the Sensor Simulation
- Basic LIDAR Simulation
- Basic Camera Simulation
- Conclusions



Application- Terrain Renderer

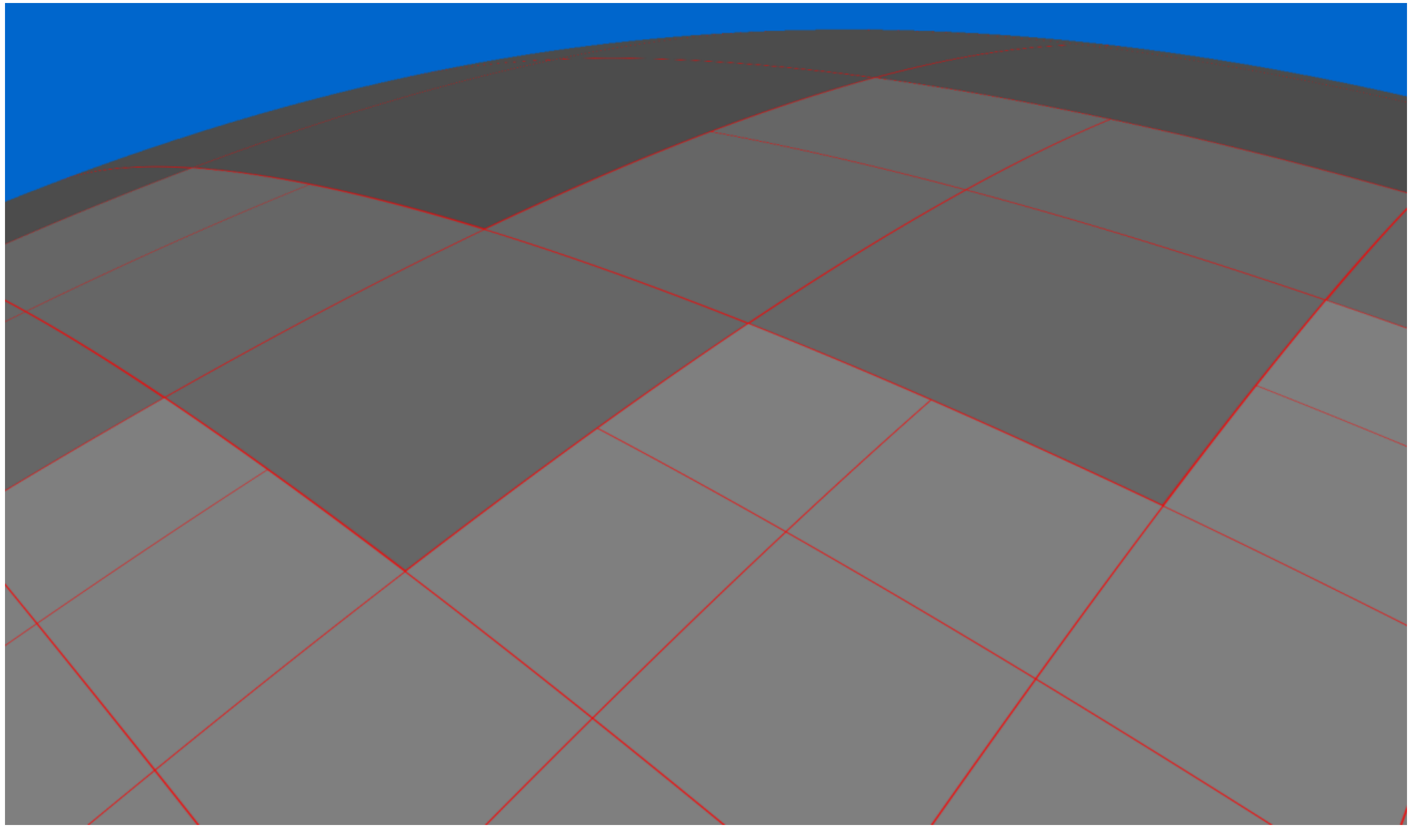
- Previously introduced in the 2011 MEON workshop
- Interactive visualization
- For a wide range of applications



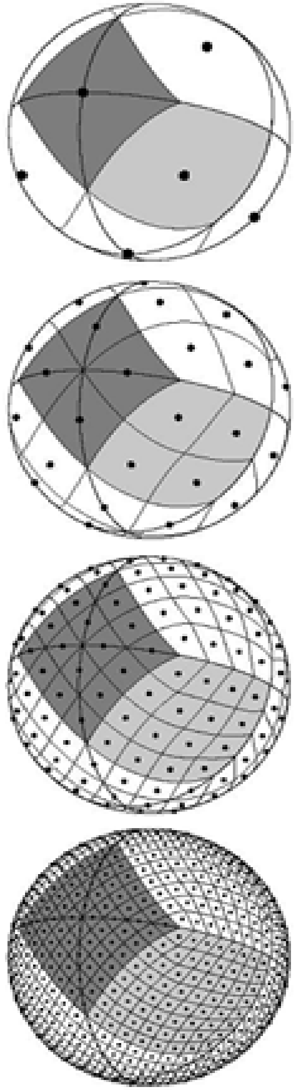


Application- Terrain Renderer

- *Level of Detail Approach (LOD)* with the HealPix (NASA) data structure.



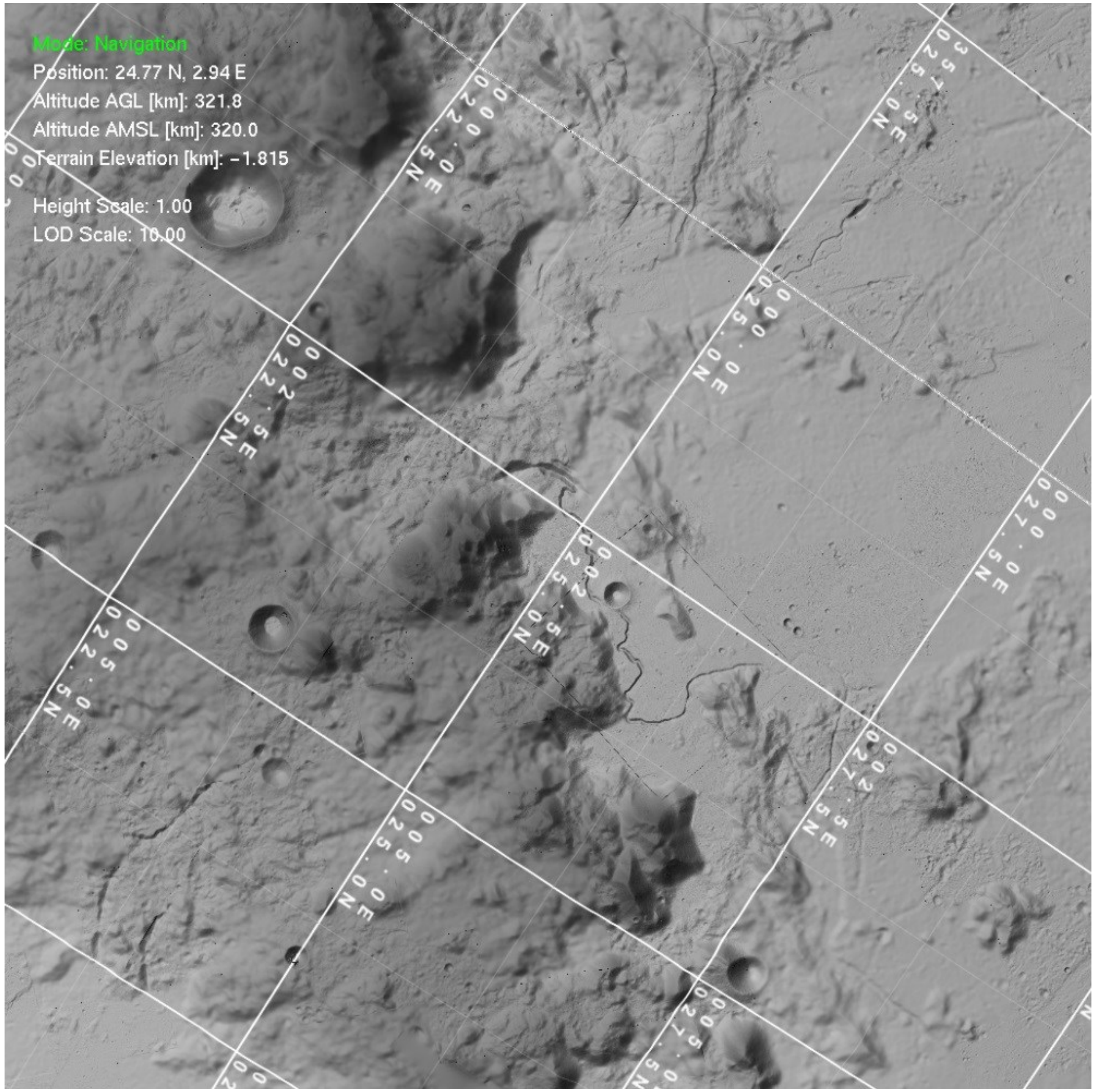
Level of Detail (LOD) Approach



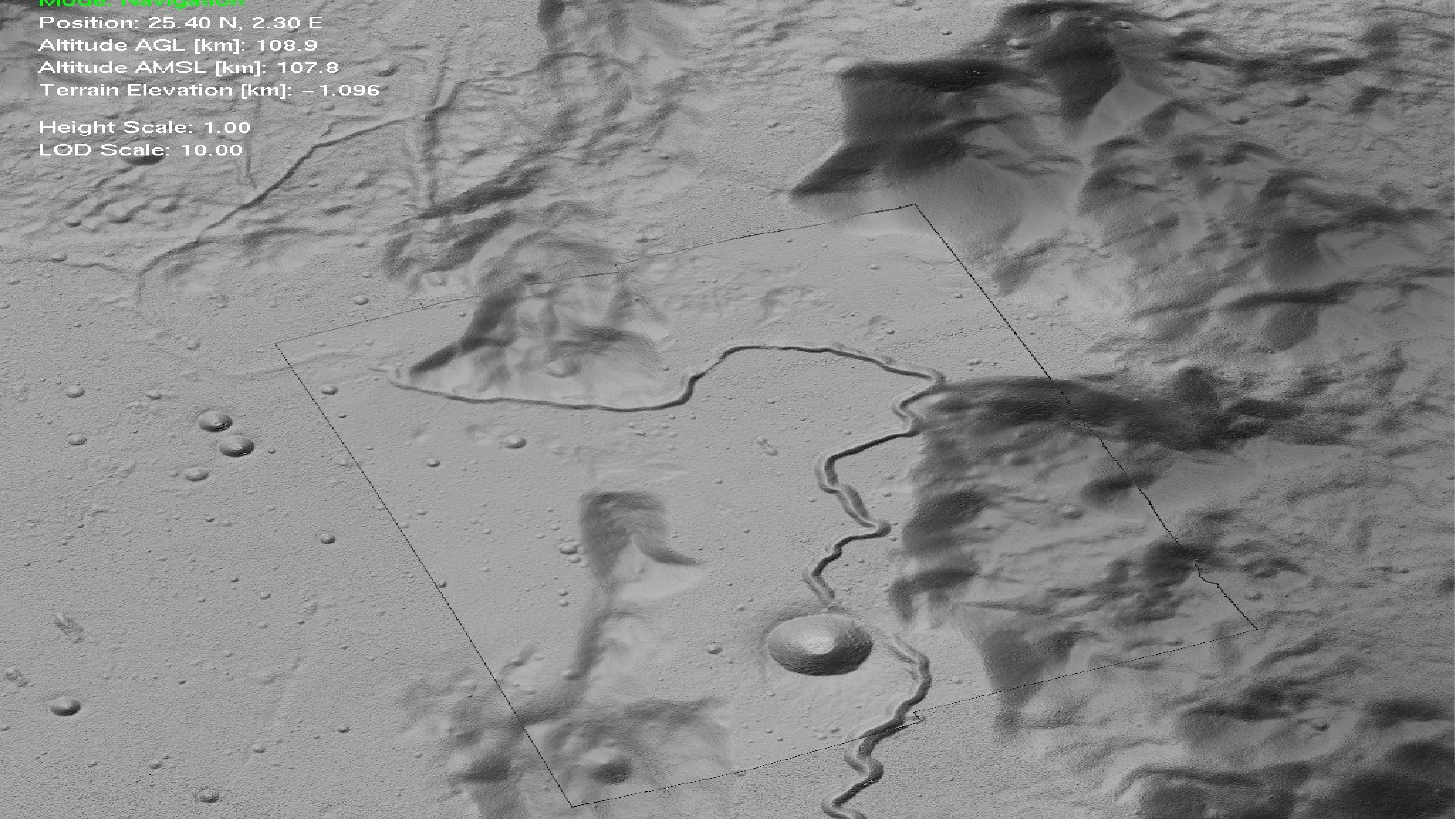
HealPix

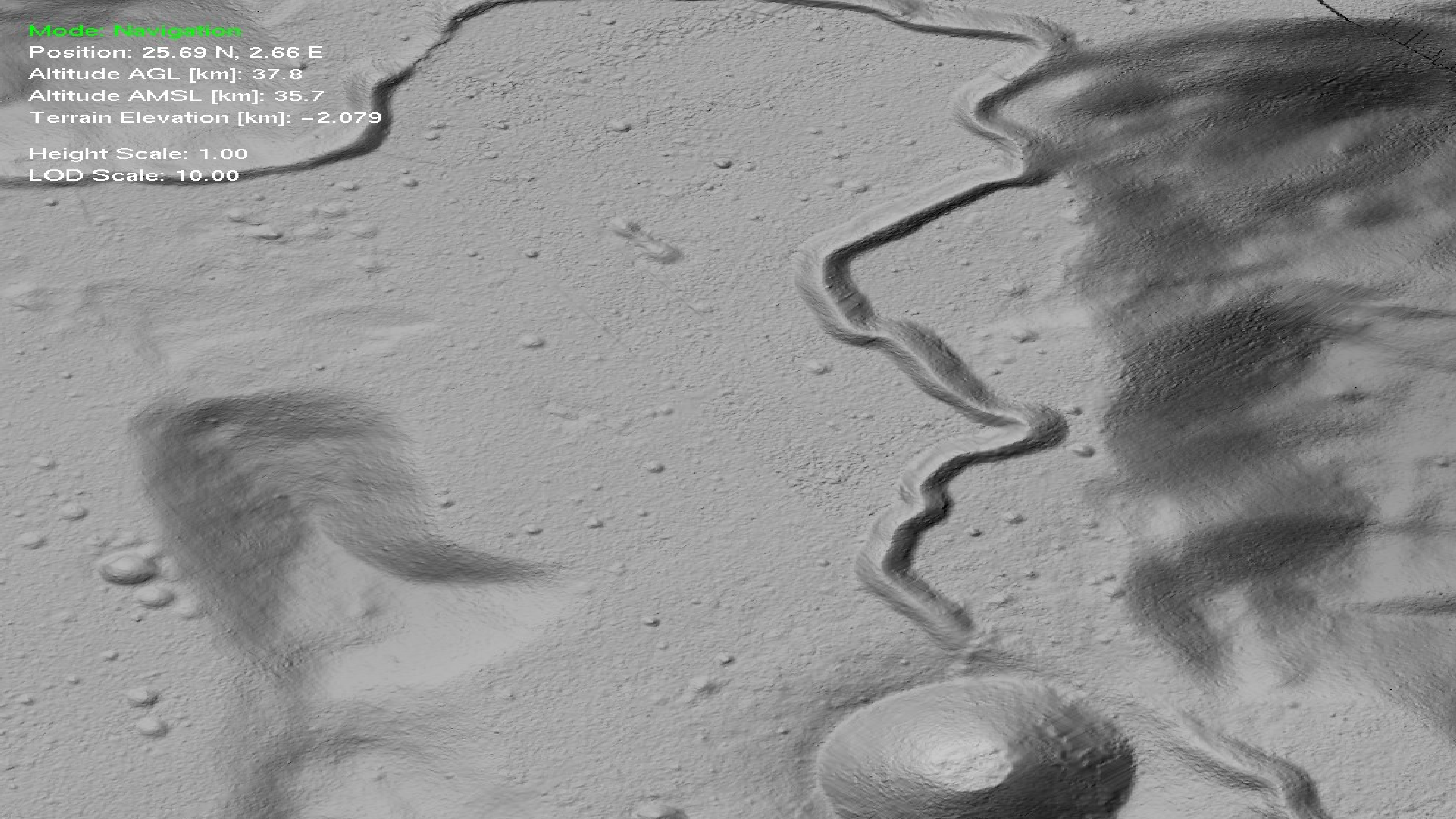
Application- Terrain Renderer

- Various datasets
- For moon KAGUYA + LRO, 4.1 TB
- Hybrid data sets. Eg. Image data

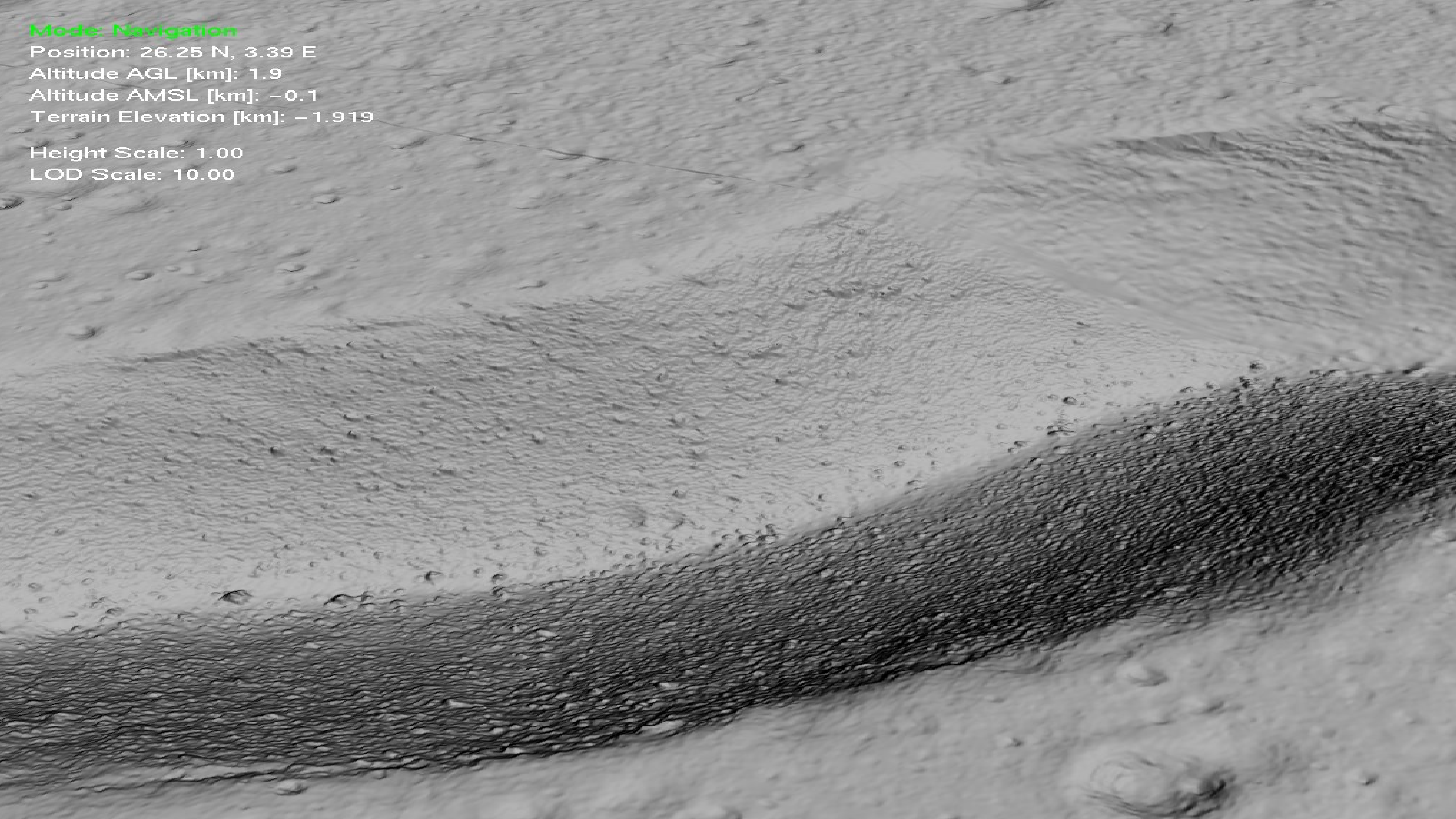


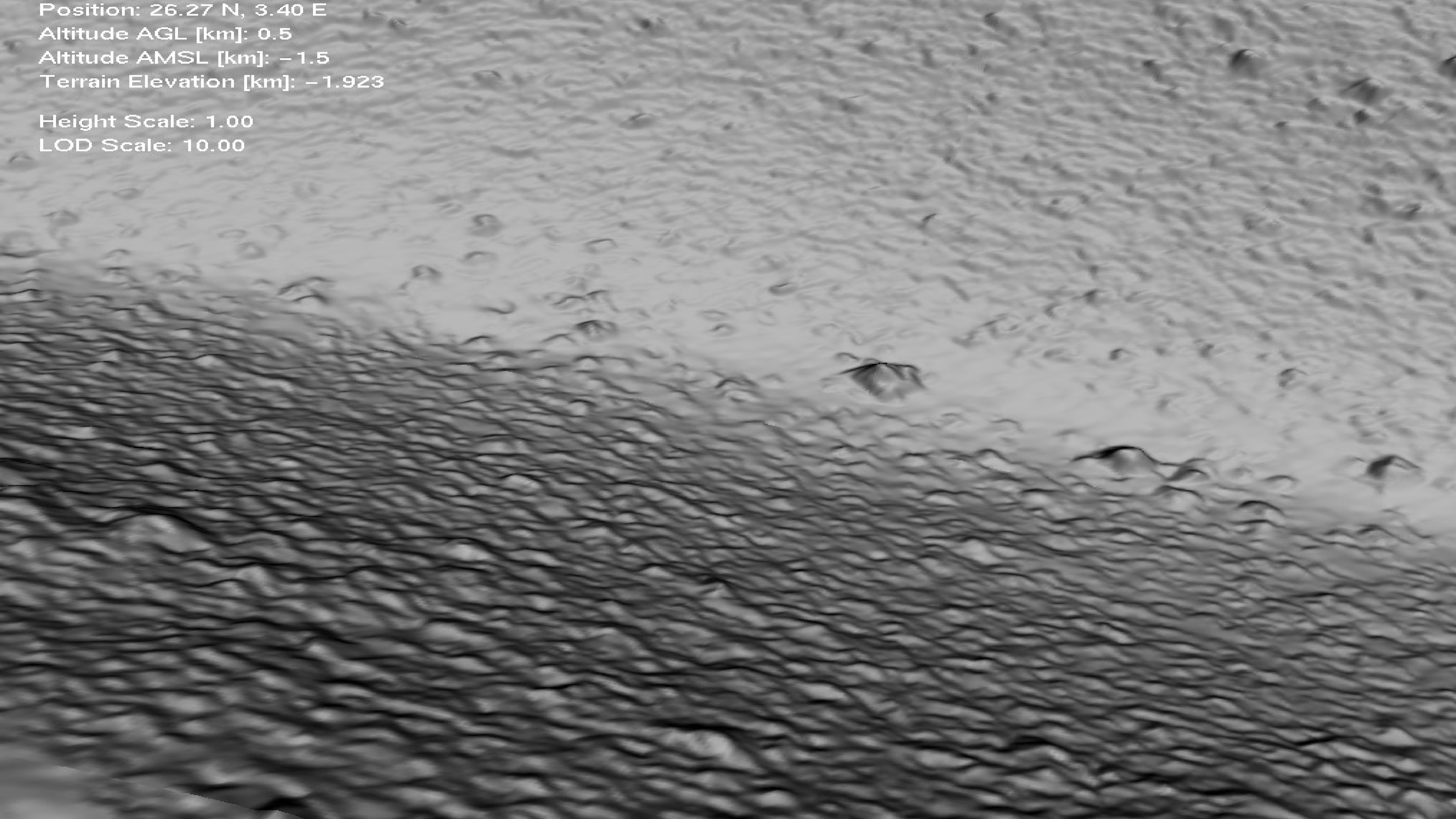
Apollo 15 Landing Area (DLR Terrain Renderer)


Position: 25.40 N, 2.30 E
Altitude AGL [km]: 108.9
Altitude AMSL [km]: 107.8
Terrain Elevation [km]: -1.096
Height Scale: 1.00
LOD Scale: 10.00


Mode: Navigation
Position: 25.69 N, 2.66 E
Altitude AGL [km]: 37.8
Altitude AMSL [km]: 35.7
Terrain Elevation [km]: -2.079
Height Scale: 1.00
LOD Scale: 10.00


Mode: Navigation
Position: 26.03 N, 3.42 E
Altitude AGL [km]: 13.1
Altitude AMSL [km]: 11.1
Terrain Elevation [km]: -1.932
Height Scale: 1.00
LOD Scale: 10.00

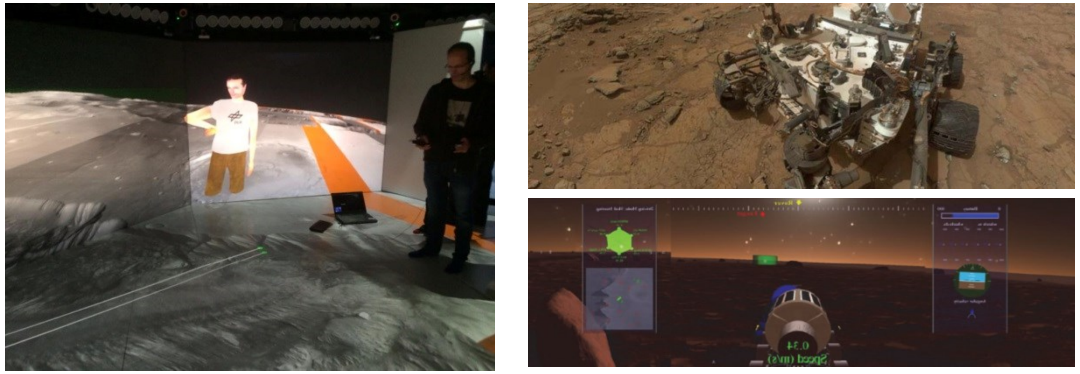

Mode: Navigation
Position: 26.25 N, 3.39 E
Altitude AGL [km]: 1.9
Altitude AMSL [km]: −0.1
Terrain Elevation [km]: −1.919
Height Scale: 1.00
LOD Scale: 10.00


Position: 26.27 N, 3.40 E
Altitude AGL [km]: 0.5
Altitude AMSL [km]: −1.5
Terrain Elevation [km]: −1.923
Height Scale: 1.00
LOD Scale: 10.00

Motivation

- High Accuracy and Real time Requirements from Space Domain for Sensor Simulation
- ATON (*Autonomous Terrain based Optical Navigation*)
- CROSS DRIVE (*Collaborative Rover Operations and Satellites Science in Distributed Remote and Interactive Virtual Environments*)

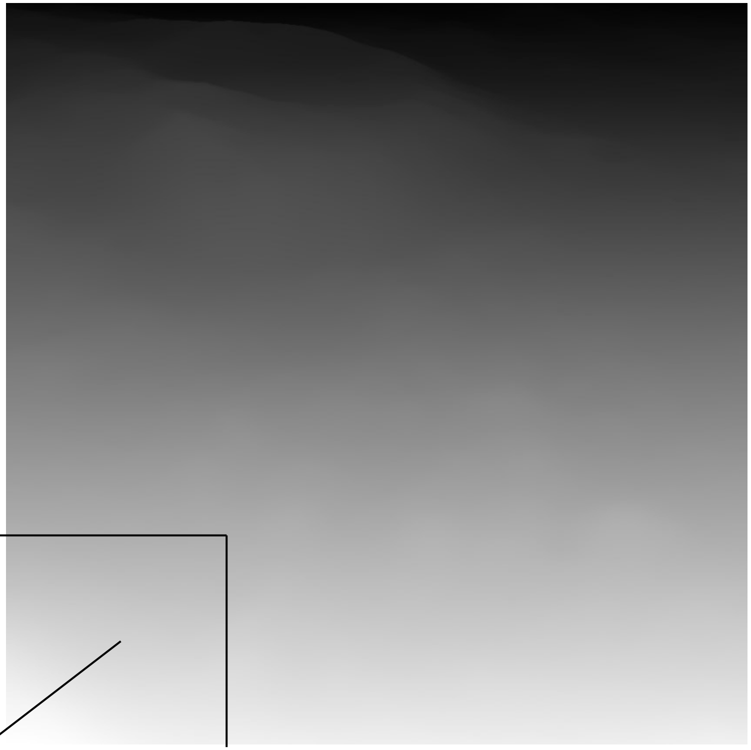
TRON (Testbed for Robotic Optical Navigation)

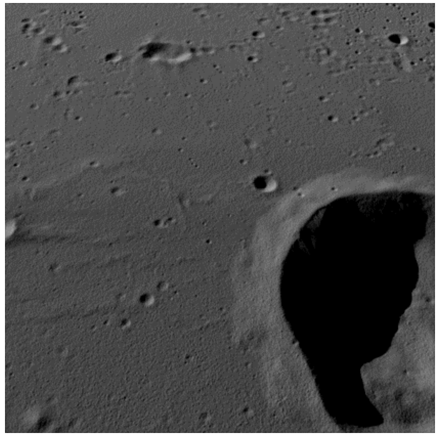


CROSS DRIVE (http://www.cross-drive.eu/)

Basic LIDAR Simulation

- LIDAR: Light Detection And Ranging
- Flash LIDAR
- Each Pixel represents a range value





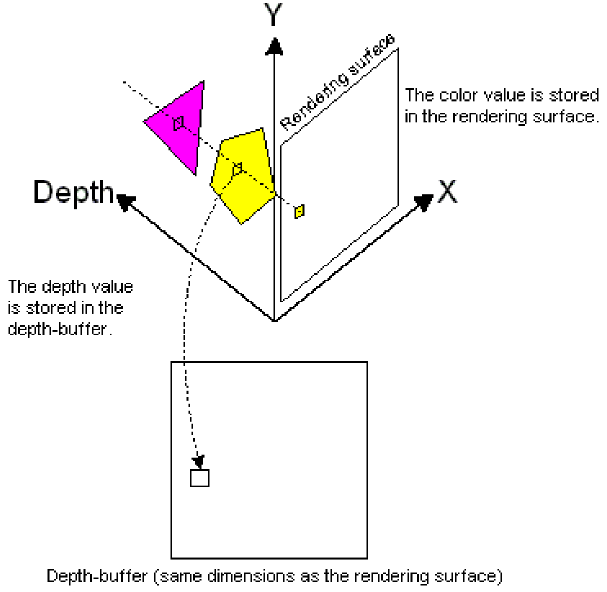
Camera Image/ DLR OS

LIDAR Image /DLR SC

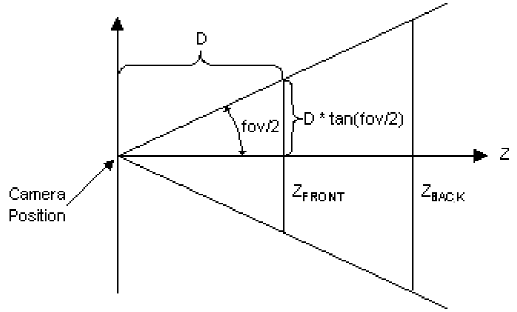
200	202	235	220	222	232	241	245	
200	222	235	220	222	232	241	224	
200	233	235	220	222	232	241	244	
200	222	234	220	222	232	240	201	
205	202	235	222	222	232	240	244	
200	202	235	220	222	232	241	243	
205	202	235	222	222	232	241	244	
205	202	235	222	222	232	240	244	

Basic LIDAR Simulation

- LIDAR Images by OpenGL depth buffer.
- Depth Buffer measures the range between 0 – 1, non-linear [4].
- Transformations and quantization applied.
- Accuracy may suffer.
- Dynamic Clipping Planes : min (Zback/ Zfront)
- Errors due to rendering and DEM.



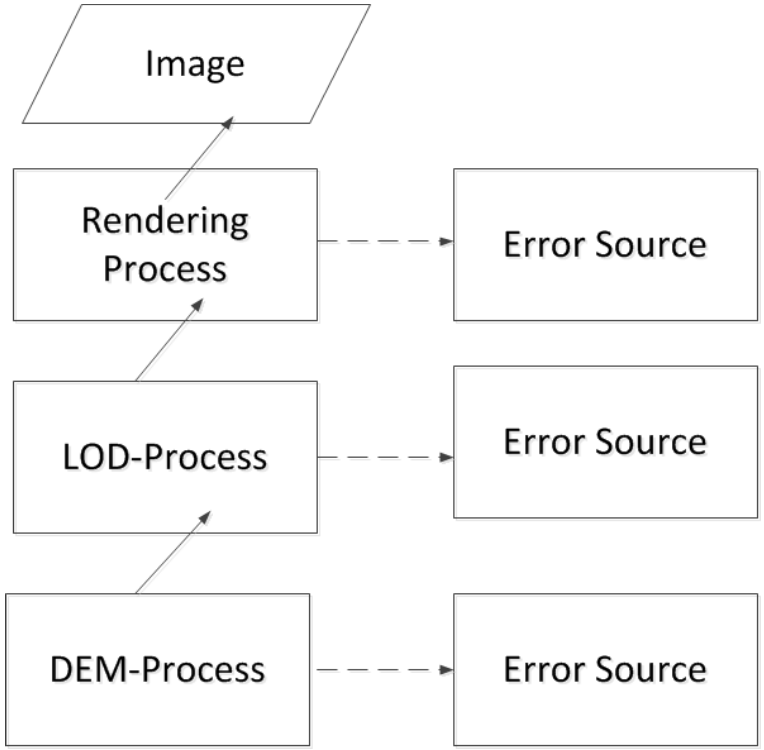
Depth Buffer



View Frustum

Basic LIDAR Simulation

- Error sources are defined.
- To detect these errors, validation software is created.
- The similar pipeline as a rendering engines follows.



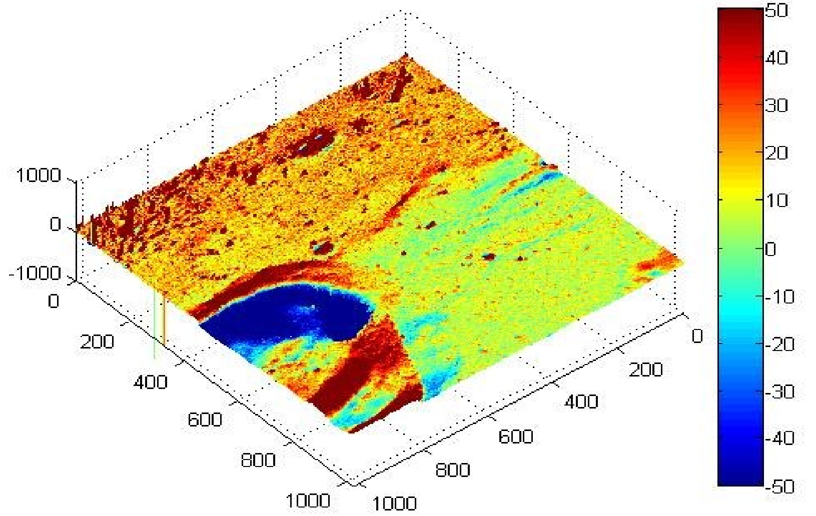
Possible Error Sources

- It determines distances from the view point to all visible triangle vertices
- It finds minimun and maximum values of each pixel

Basic LIDAR Simulation

- We tested our application :
  - Results: OpenGL Buf with dynamic clipping plane ranges works fine in the terrain renderer sofware.
  - All the pixel values are between the limits (min and max value).
  - No errors : Deviation is around %0.10 for the given figure.

Simulated LIDAR Image



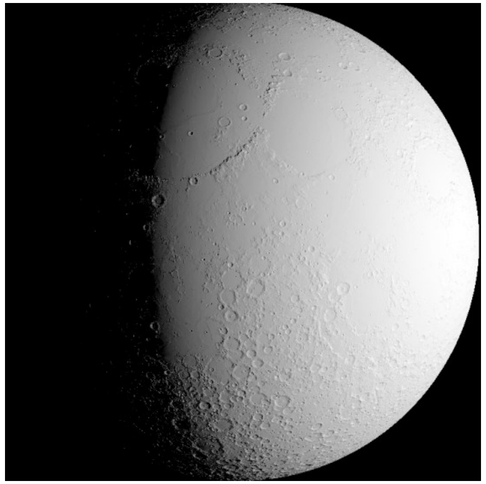
Deviation of the LIDAR Image (Units in Meters)

Basic Camera Simulation

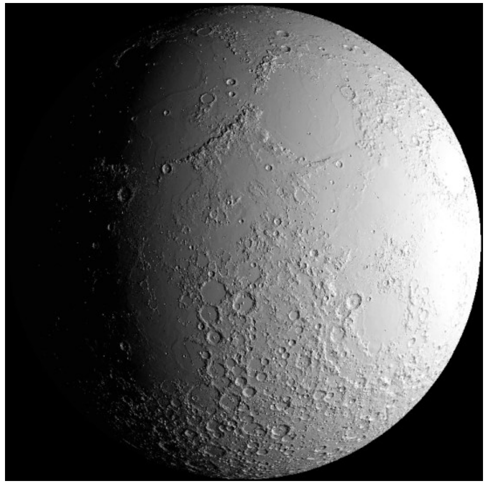
- Pin hole camera model – Basic
- To visualize correct topology of the planets and lighting
- Real time and accuracy requirements
- More challenging than LIDAR :
    - Correct lighting :
        - Correct Light Source Simulation
        - Correct Shadow Simulation

Basic Camera Simulation

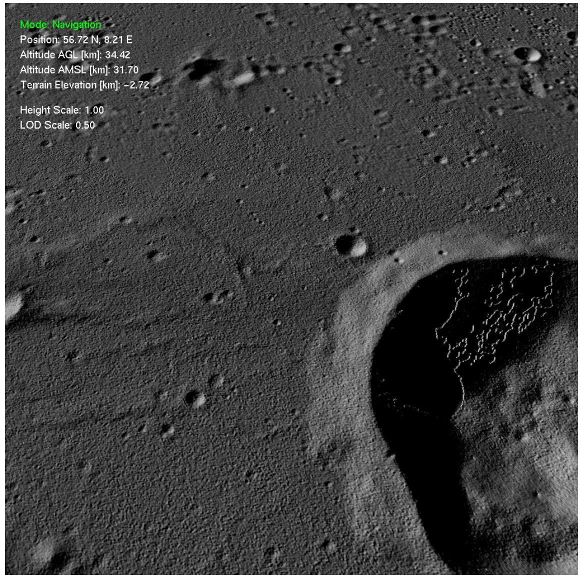
- Light Source Computation: SPICE (NASA) tool
  - Sub solar point compared with LTVT software
- The basic shading, only considers normal mapping
- Terrain Renderer lacks of accurate shadows



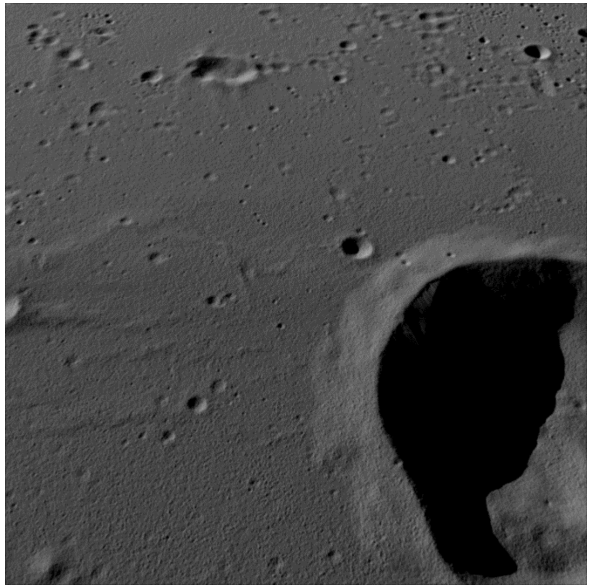
Camera Image/LTVT



Camera Image/DLR Braunschweig



Camera Image/ DLR SC

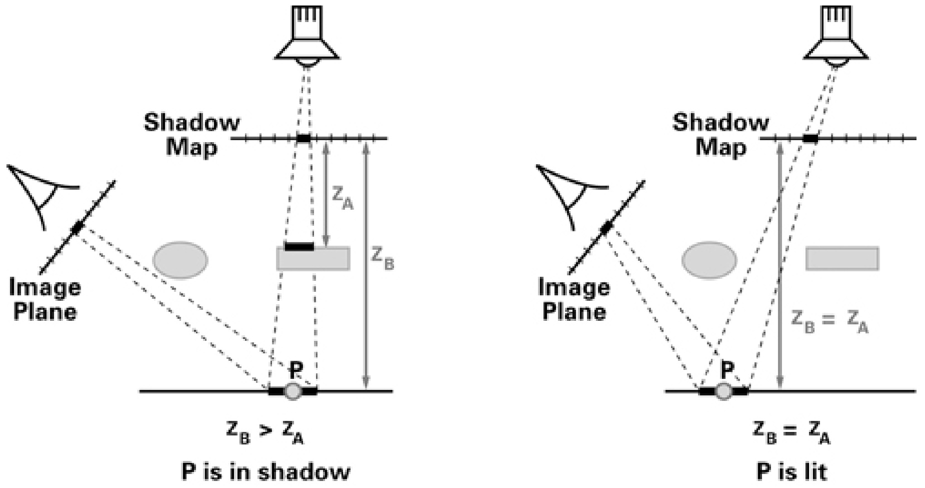


Camera Image/ DLR OS

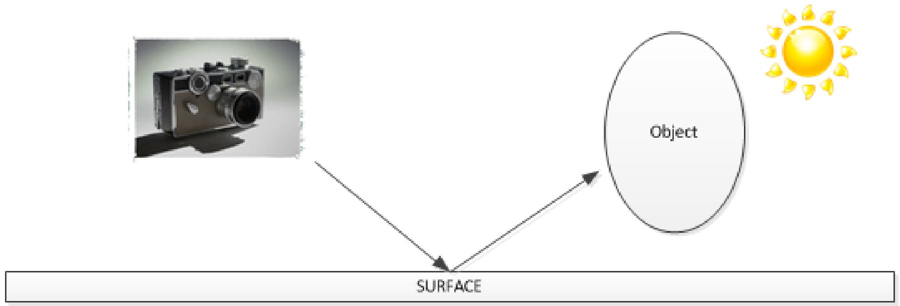
Basic Camera Simulation

- Accuracy vs Performance
- Commonly used Algorithms for shadows :
    - Raycasting , Shadow Mapping
    - Horizon Mapping for bump mapped surfaces [2]

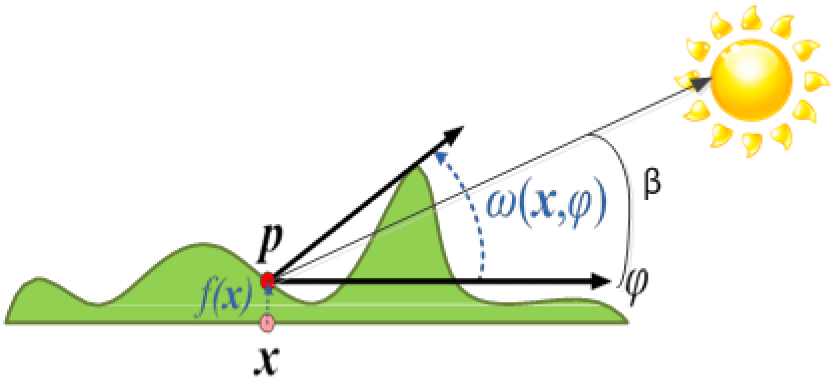
- Raycasting is computationally expensive.
- Shadowmapping renders scene from 2 points of view.
- Horizon mapping employs precomputed values.



b) Shadowmapping

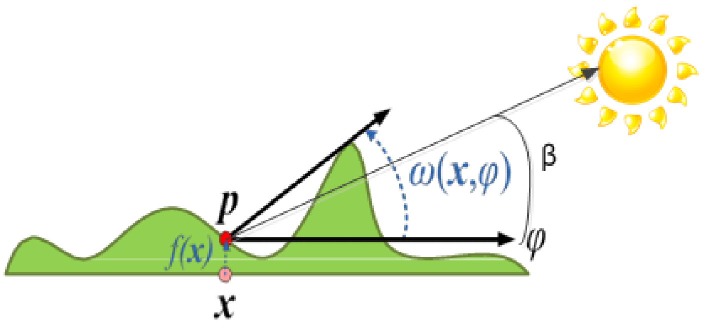


a) Raycasting

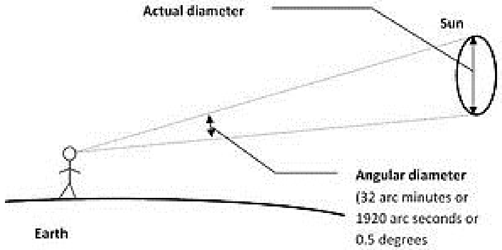


c) Horizon Mapping

Basic Camera Simulation

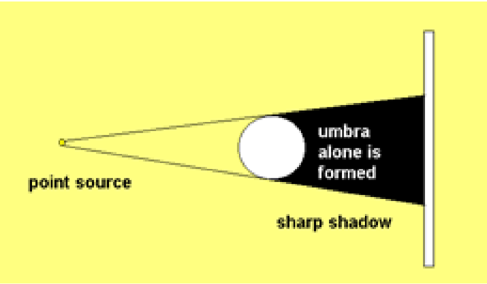


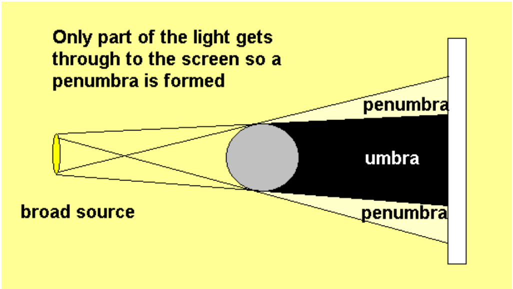
- More about Horizon Mapping
  - Algorithm: If sun is a point light source in the sky, ω - β > 0 (true:in shadow) (See Figure)



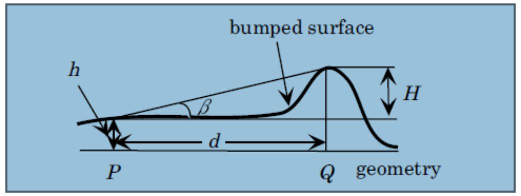
  - Sun is not a point source [1] : If δ is the angular diameter of the sun, let α = (w – β)/(δ/2) By plane geometry area formulas [2], the visible fraction of the Sun :

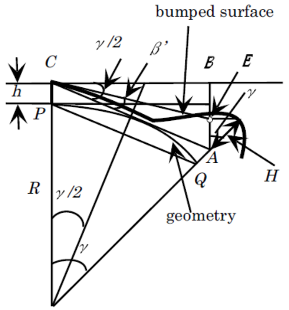
$$f(\alpha)=\begin{cases}0.0 & \alpha\leq -1\\ 0.5+(\sin^{-1}\alpha+\alpha\sqrt{1-\alpha^2})/\pi & -1\leq\alpha\leq 1\\ 1.0 & 1\leq\alpha.\end{cases}$$





  - Planets are not flat [3]

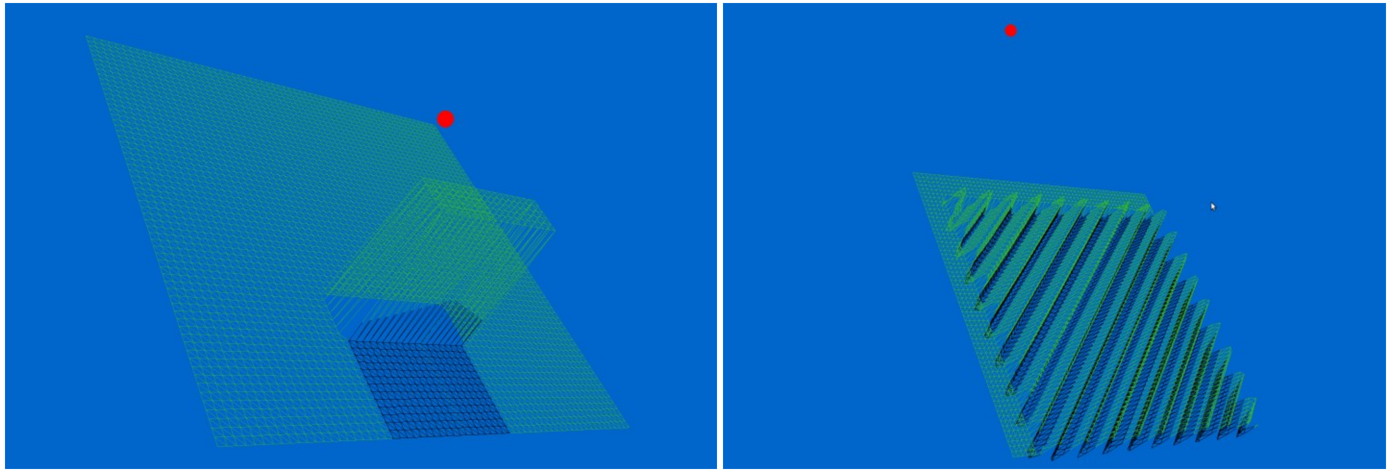




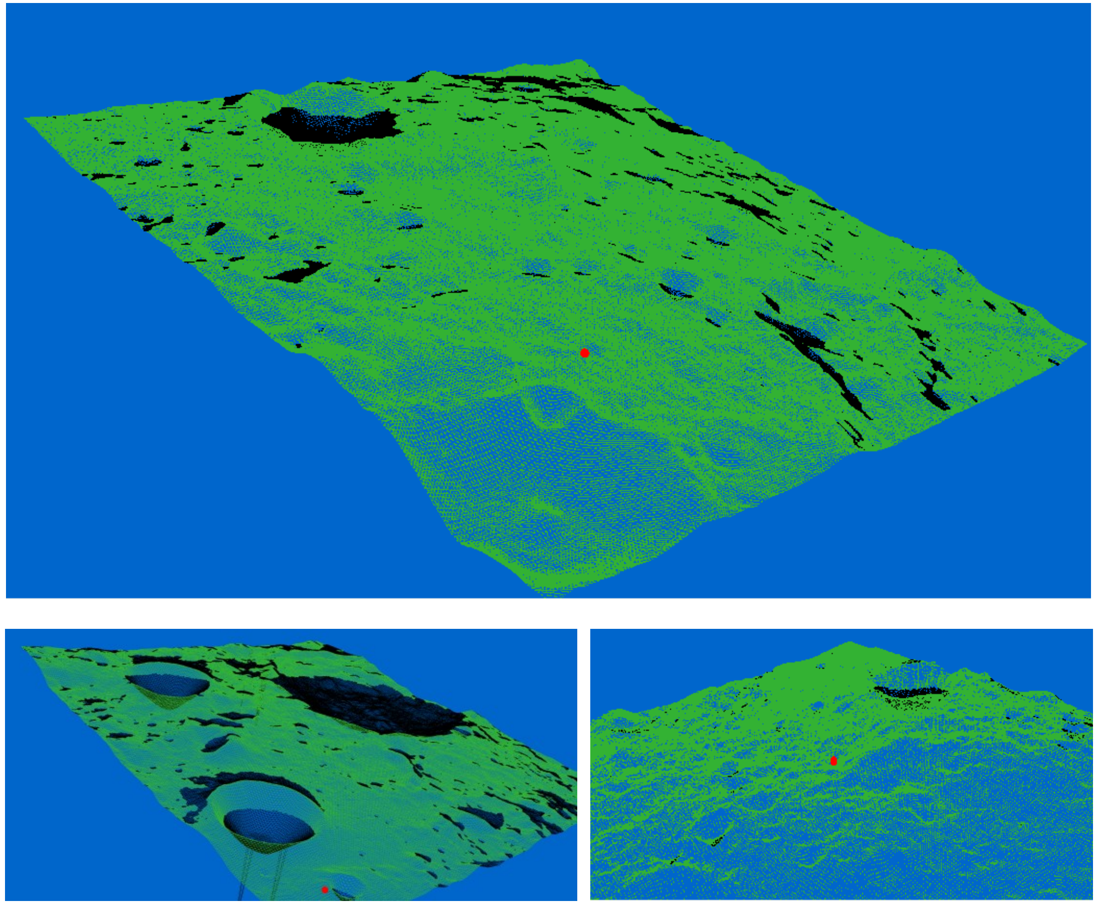
$$\beta' \approx arctan\left(\frac{d\tan\beta-2(R+h)\frac{\gamma^2}{4}}{2(R+h)\frac{\gamma}{2}}\right)$$

Basic Camera Simulation

- A prototype for *Horizon Mapping is* created.
- The accuracy analysis for the camera simulation is in progress.



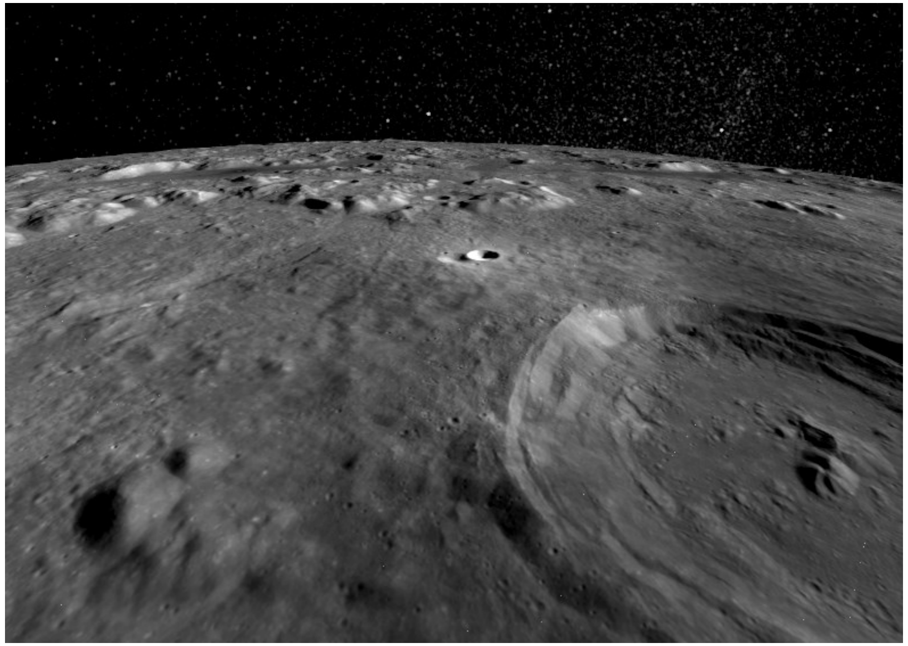
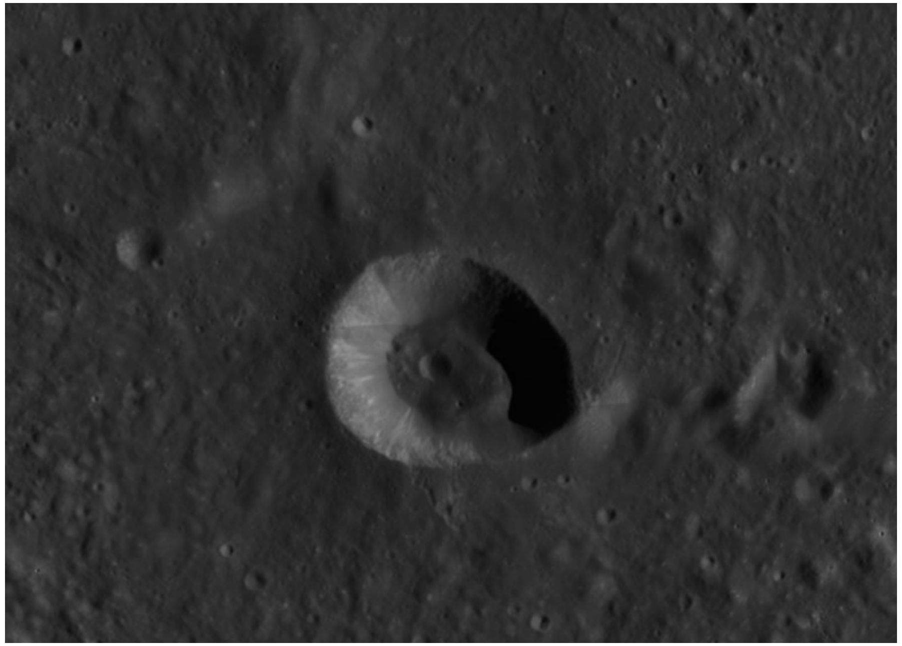
Results from Our Prototype with an Artifical Mesh



Results from Our Prototype with Kaguya Dataset

Basic Camera Simulation

- Next steps:
  - Integration of the algorithms to the application.
  - Evaluation of the shadows by comparing the real images of Moon with the artificial ones.

DLR Terrain Renderer, the images (taken by NASA) are mapped to Moon surface.

Conclusions

- An overview of the terrain renderer application is given.
- LIDAR simulation is presented. The possible error sources demonstrated and the validation is descibed.
- The state of our work is given for the camera simulation. The work is still in progress. We are still evaluating shadowing algorithms for our application.
- The terrain renderer application is on its way to give both real time (See our demo) and accurate results for the sensor simulation.

Thanks

Turgay Aslandere
German Aerospace Center
Simulation and Software Technology | Software for Space Systems and Interactive Visualization
Lilienthalplatz 7 | 38108 Braunschweig | Germany
Telephone +49 531 295 2956

Image: DLR Terrain Renderer, Moon North Pole 87 N, 28 E

References

[1] Horizon Mapping for bump-mapped surface, Nelson L. Max, 1988

[2] Interactive Horizon Mapping Peter Pike, Micheal Cohen, 2000

[3] Real-time Rendering of Bump map Shadows Taking Account of Surface Curvature Koichi Onoue, Nelson Max, 2004

[4] http://msdn.microsoft.com

[5] http://ltvt.wikispaces.com/LTVT

[6] https://developer.nvidia.com/

DLR