PANGU v7

Maintenance and Upgrade of PANGU (Planet and Asteroid Natural Scene Generation Utility)

Executive Summary

Ref: UoD-PANGU-v7-ES

Revision: Issue 1.0

Date: 26th November 2023

Space Technology Centre
University of Dundee
166 Nethergate, Dundee, DD1 4EE, Scotland, UK

Document Authors

Iain Martin

Martin Dunstan

Document Change Log

Date	Revision No	Comments
26th November 2023	Issue 1.0	First issue

A comprehensive list of the changes made to this document in each revision is in Section 7.

CONTENTS

I LIST OF FIGURES

II ACRONYMS AND ABBREVIATIONS

AD	Applicable Document
BRDF	Bidirectional Reflectance Distribution Function
CAD	Computer Aided Design
CPU	Central Processing Unit (PC)
DEM	Digital Elevation Model
GNC	Guidance, Navigation and Control
GPU	Graphics Processing Unit (graphics card)
LUT	Look Up Table
NCR	Non Conformance Report
NRB	NCR Review Board
OBJ	Wavefront|OBJ file format for storing simple geometry
PDS	Planetary Data System (https://pds.jpl.nasa.gov)
PGM	Portable Grey Map (NetPBM grey scale image format)
PNG	Portable Network Graphics (image file format)
PPM	Portable Pixel Map (NetPBM colour image format)
PSF	Point Spread Function
RD	Reference Document
RGB	Red, Green, Blue (a tri-stimulus colour model)
SoW	Statement of Work
SPR	Software Problem Report
TIR	Thermal infrared part of the electromagnetic spectrum
UoD	University of Dundee
VIS	Visible part of the electromagnetic spectrum

1 INTRODUCTION

The overall aim of this activity was to maintain and extend the existing PANGU planet and asteroid simulation tool so that it can offer improved support to verify and validate vision-based navigation systems throughout the design and qualification phases of ESA space programmes using real-time avionics test benches and other approaches.

PANGU v7 enhancement and maintenance had the following main goals which were split into three categories and were all achieved:

- Category 1: Enhancement with new features:
  - Improvements on the implementation of the synthetic image generation in the thermal IR for planets/asteroids and S/Cs.
  - Improvement of simulations on the surface of planets and moons for rovers.
  - Versatility on camera and sensor models.
- Category 2: Maintenance of the SW (correction of bugs):
  - Corrections of the bugs reported as SPRs and NCRs.
  - General maintenance tasks.
- Category 3: Maintenance and upgrade of the SW and the PANGU community portal:
  - Web portal for PANGU with forums, wiki, problem reporting, announcements, etc.

The category 1 tasks were split into developing new features and scenario examples to enhance PANGU in the following areas:

- High-fidelity rendering of terrain imaging (Moon and Mars and Asteroids),
- Enhanced thermal imaging for terrain (Moon and Mars and Asteroids) and spacecraft, and
- Sensor simulator for Processor in the Loop of GNC systems.

These are described in more detail in the following sections.

2 HIGH-FIDELITY RENDERING OF TERRAIN IMAGING

New features developed to enhance high-fidelity rendering of terrain imaging included feature-related albedo modifications for dunes and craters, high-frequency texture filtering on terrain models and an experimental method to import multi-lobed OBJ objects (asteroid shape models). In addition, two new example lunar scenarios were developed to provide representative lunar simulations from large and high-resolution DEMs.

2.1 FEATURE-RELATED ALBEDO MODIFICATIONS

PANGU v6 supported global albedo maps for small-bodies and DEM-based models. However, some terrain features, such as dark sand dunes on Mars, may show clear a correlation between terrain features and related albedo. The main feature-related albedo effects identified for PANGU were dune fields, and craters with bright or dark halos. User-controlled dune albedo modifications have been implemented to blend albedo variations into the main albedo map. A height range can be defined over which to blend the albedo variations into the albedo map, to avoid unrealistically abrupt albedo changes at the dune field boundary. For craters, albedo variations can be applied in radial regions from the crater centre to simulate the effect of dark or bright halos, with user parameters provided to control the albedo modifications. An example of PANGU feature-related albedo is shown in Figure 1 with darkened Barchan sand dunes on lighter coloured terrain and is compared to a real MRO image showing dunes darker than the surrounding terrain.

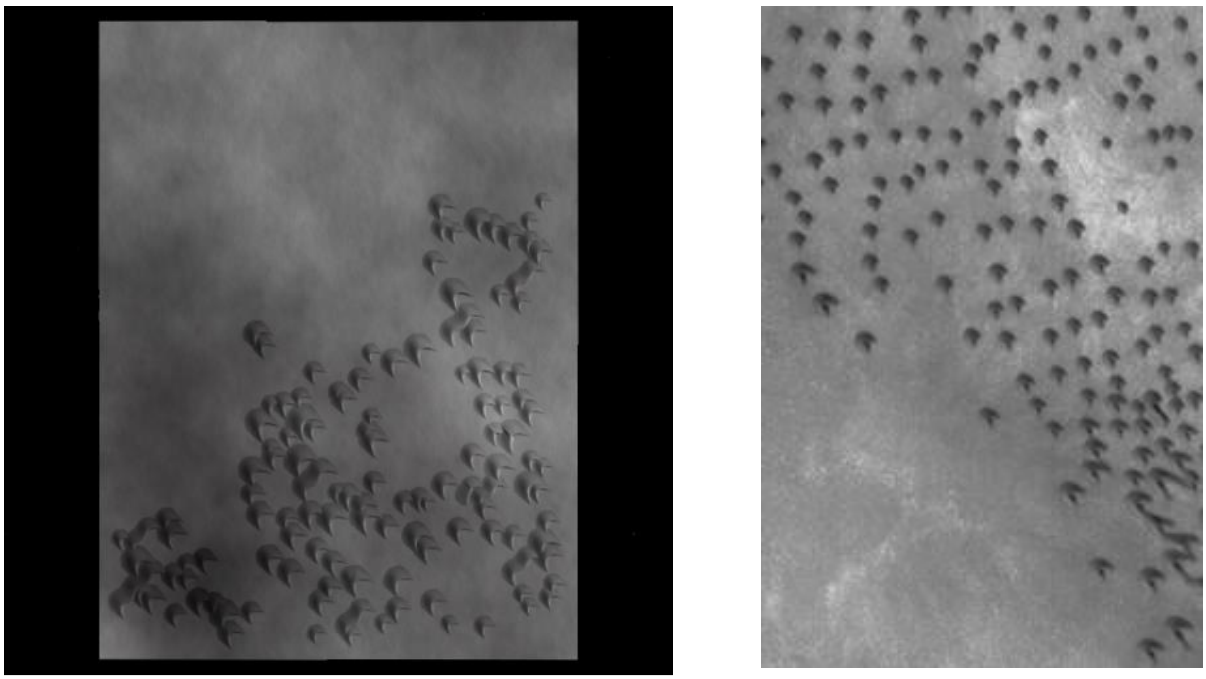
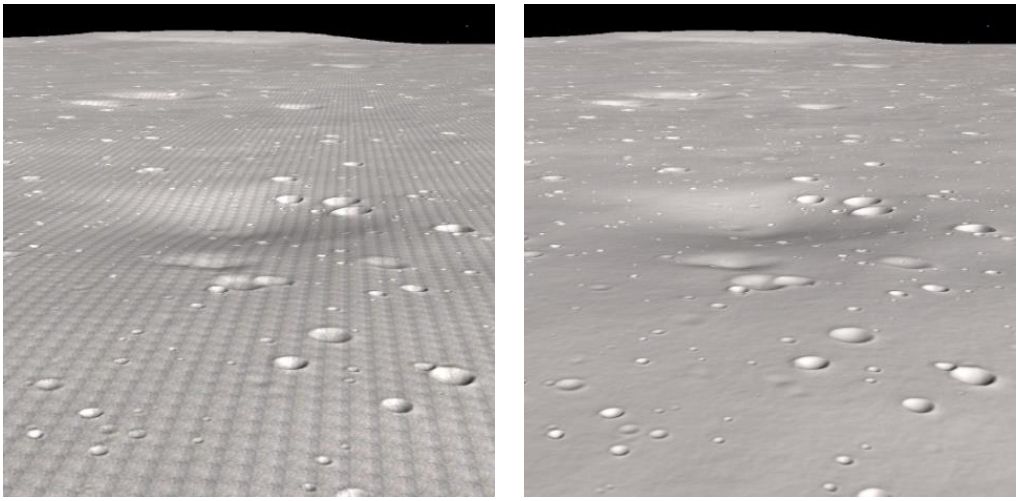


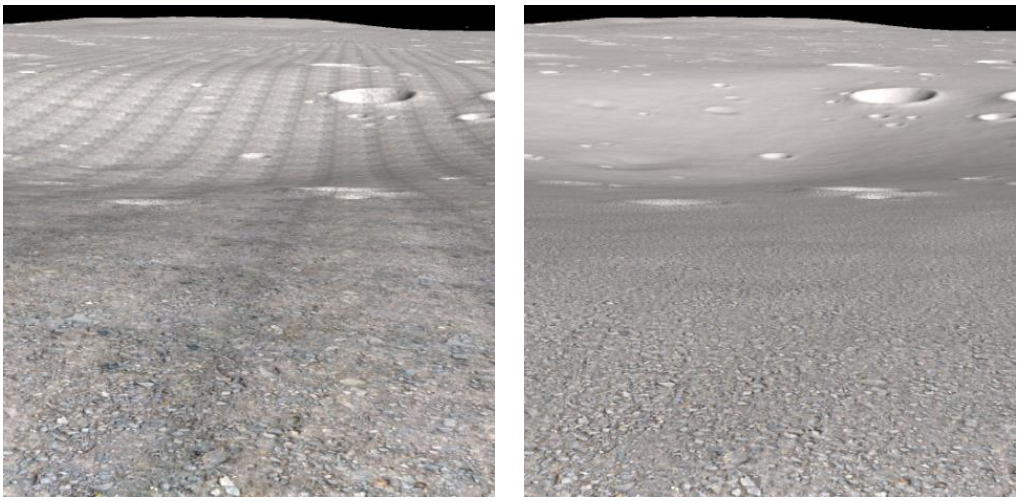
Figure 1: PANGU feature-related albedo (left), MRO HiRISE image PIA05920 (right)

2.2 TEXTURE FILTERING

Texturing in PANGU utilises the mip-mapping technique to provide appropriately filtered texture at all camera view distances. A mip-map can be considered to be a pyramidal stack of textures in which each successive level has half the resolution as the previous layer *e.g.* 256×256, 128×128,…,1×1. Each lower resolution level is obtained by averaging each group of 2×2 pixels from the adjacent higher resolution level (known as box filtering). The rendering system selects the level that is most suited for the current view, blending between levels to provide smooth changes with distances. This system is effective at avoiding aliasing that would result from attempting to render a texture at much higher resolution than the output image can support.



Camera distance = 350m Camera distance = 350m



Camera distance = 20m Camera distance = 20m

Figure 2: Rendering without (left) and with (right) texture filtering

To avoid the gridding effect at large distances, the lowest frequency terms of the texture must be removed. This can be achieved by ensuring that the lower resolution levels of the mip-map pyramid (corresponding to the largest camera distances) are set to a constant value. A texture with dimensions $2^n \times 2^n$ will contain n+1 levels; level 0 is the highest resolution (the original image) while level n is the lowest resolution 1×1 i.e. holding the DC (average) value. Level n-k will hold all the lowest k non-DC frequency terms of the texture. Therefore, if the value of each pixel of level n-k (minus the DC term from level n), is subtracted from all the pixels of level 0 covered by that level n-k pixel, the lowest k non-DC frequency terms will be eliminated from the texture. Texture filtering to remove low-frequency components can be applied to surface textures for visual and thermal rendering images. An example of texture filtering to remove gridded artefacts is shown in Figure 2.

2.3 LUNAR CASE STUDY EXAMPLES

PANGU v6 includes two large lunar model examples, a whole Moon model and a South Pole model, landing on Malapert peak. These were created with the best available DEMs at that time. However, larger and more accurate lunar DEMs are now available so two new lunar scenarios have been developed to better support the current lunar exploration activities: a low-orbit descent simulation requiring simulation coverage of a large section of the Moon based on a combined LRO/Kaguya DEM, and a landing-site simulation requiring a very high-resolution model to simulate the final stages of guidance and hazard avoidance, based on an LROC NAC high-resolution DEM. These new examples demonstrate importing and inspecting a DEM, creating a PANGU-enhanced high-resolution model with additional features (*e.g.* craters and boulders) added below the resolution of the base DEM, assigning BRDF material properties and surface texture, and running simulations. Their inclusion has been designed to simplify the process for users wishing to run comparable simulations with different base DEMs and mission scenarios. Users can follow the processes used to run these simulations and adapt them to their own scenario as required, including using different DEMs with similar properties.

2.3.1 LROC NAC high-resolution example

For this scenario, the high-resolution LROC stereo DEM covering (4.81172093N, 338.50996587E) to (3.05020714S, 338.97092643E) was used which happens to include the Apollo 12 Ascent Stage impact site. It has 2790×10683 32-bit samples, a grid resolution of 5m, a precision error of 1.95m and uses equirectangular (cylindrical) projection. This is a “typical” LRO NAC DEM which contains unknown (hole) values around the outside of the DEM. In this scenario, PANGU is instructed to map all invalid values to the minimum valid height of the DEM: a new feature for v7. This leaves flat terrain around the edges and below the height of most of the terrain, to be unintrusive to the simulation.

A grey-scale height map of the DEM is shown in Figure 3, beside a PANGU image rendered from the imported DEM, which also shows the high-resolution region overlaid. A high-frequency DEM overlay is used to add roughness to the imported DEM: a synthetic DEM with same dimensions as the input DEM is generated with a small height range using Perlin Noise. Adding a high frequency overlay can be useful if the input DEM is unrealistically smooth *e.g.* as a result of storing low resolution data in a high resolution DEM, or if noise reduction has been applied.

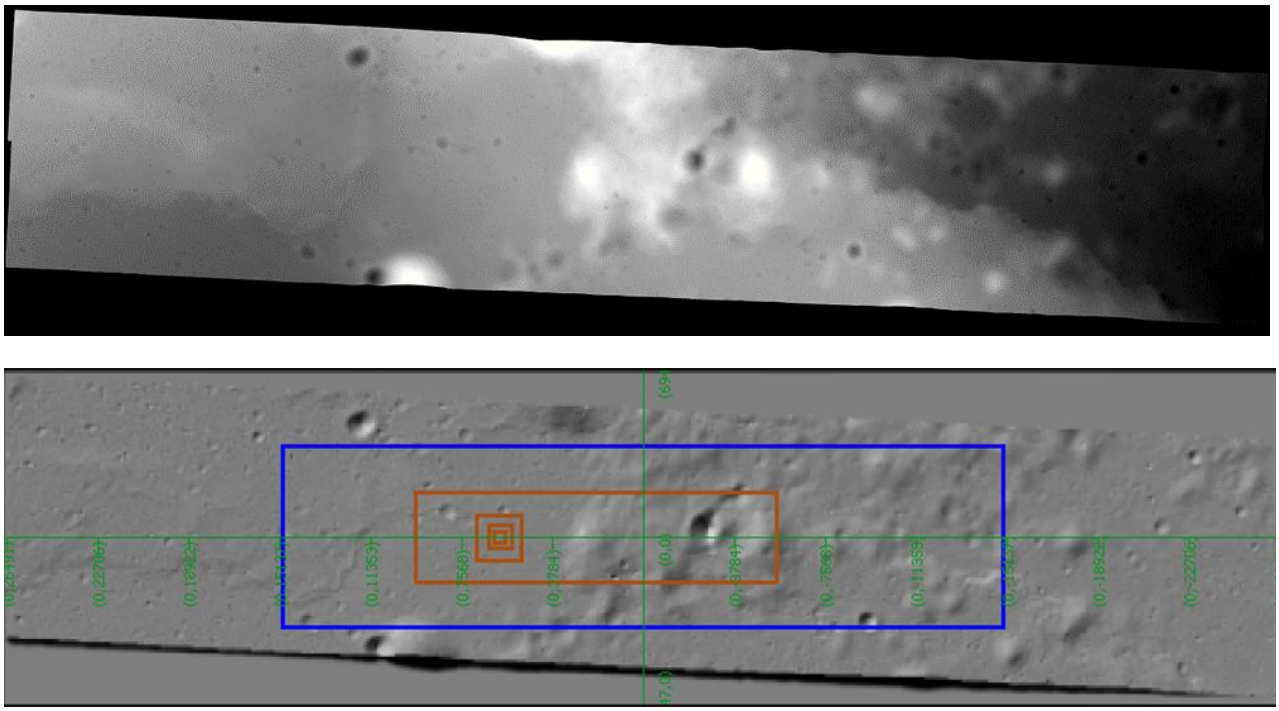


Figure 3: LROC NAC DEM (top) imported and rendered in PANGU (bottom)

The model has been synthetically enhanced by PANGU in the region shown in Figure 3. The enhancements include a synthetic albedo map, craters and boulders below the resolution of those defined in the input DEM, the high-resolution region and a filtered texture. With a horizontal resolution of ~5 m, craters below ~15–20m will not be clearly defined, so a standard lunar crater distribution was applied to the higher resolution region for craters less than 20m in diameter. An example flight file defining a trajectory flying from east to west along the centre of the DEM has been included. A selection of images from this simulation is shown in Figure 4. Features in the images from camera distances below 2000m are mostly synthetic while the 20m and low camera angles images show the use of a rough Earth gravel-type texture. Texture filtering is used to avoid gridding artefacts in the longer distance views.

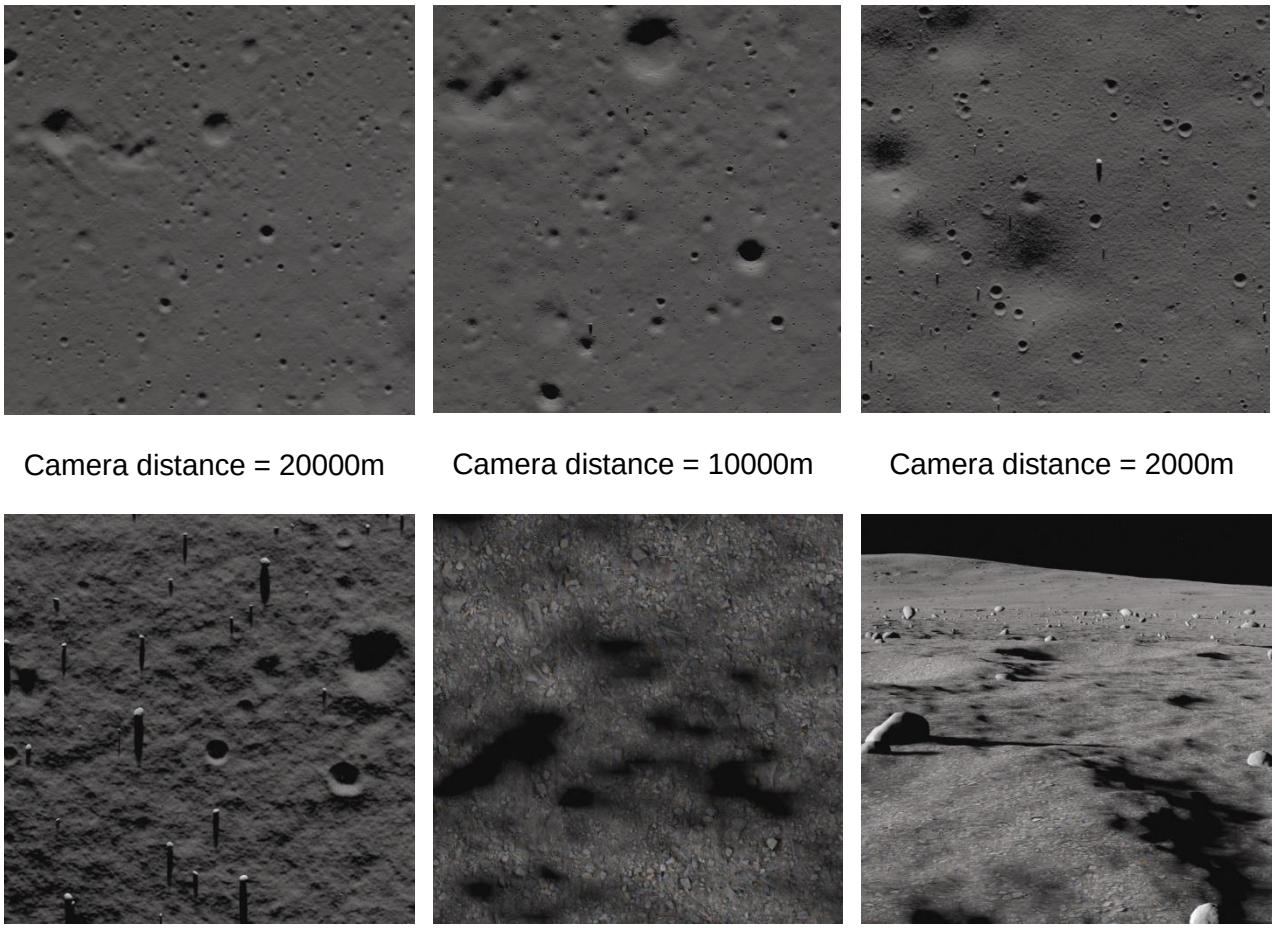


Figure 4: A selection of PANGU-generated LROC/NAC descent images

2.3.2 SLDEM low-altitude descent example

The combined LRO/Kayguya SLDEM was identified as the most appropriate DEM to use for this example because it is available at different resolutions for the whole lunar surface. The version we selected was `SLDEM2015_128_60S_60N_000_360_FLOAT.IMG` which covers latitudes 60S to 60N at a resolution of 128 pixels/degree which is ~237m horizontal resolution at the equator. The DEM is 46080×15360 in dimension, with single-precision 32-bit floating-point samples in PC_REAL format, and a height range of -8717m to 10778m.

The approach to create this low-altitude descent scenario based on this DEM was as follows. First, we imported the SLDEM into PANGU at different resolutions to allow tests on lower powered PCs as well as high-end ones. We then defined a simple trajectory flying on a great circle over the middle of the DEM in a PANGU flight-file, and the PANGU regions file (multi-resolution model) to specify a higher-resolution landing site towards the end of that trajectory to ensure that there was sufficient

model resolution at the end of the flight to render high-resolution images from. Craters are added to the higher resolution regions using a standard lunar size-density distribution. The maximum diameter is limited to that of the smallest crater visible in the input base DEM. Finally, we define camera model parameters including radiation effects, and generate images from the descent sequence defined in the flight-file. A selection of example images from this simulation are shown in Figure 5, with the yellow line showing the flight path.

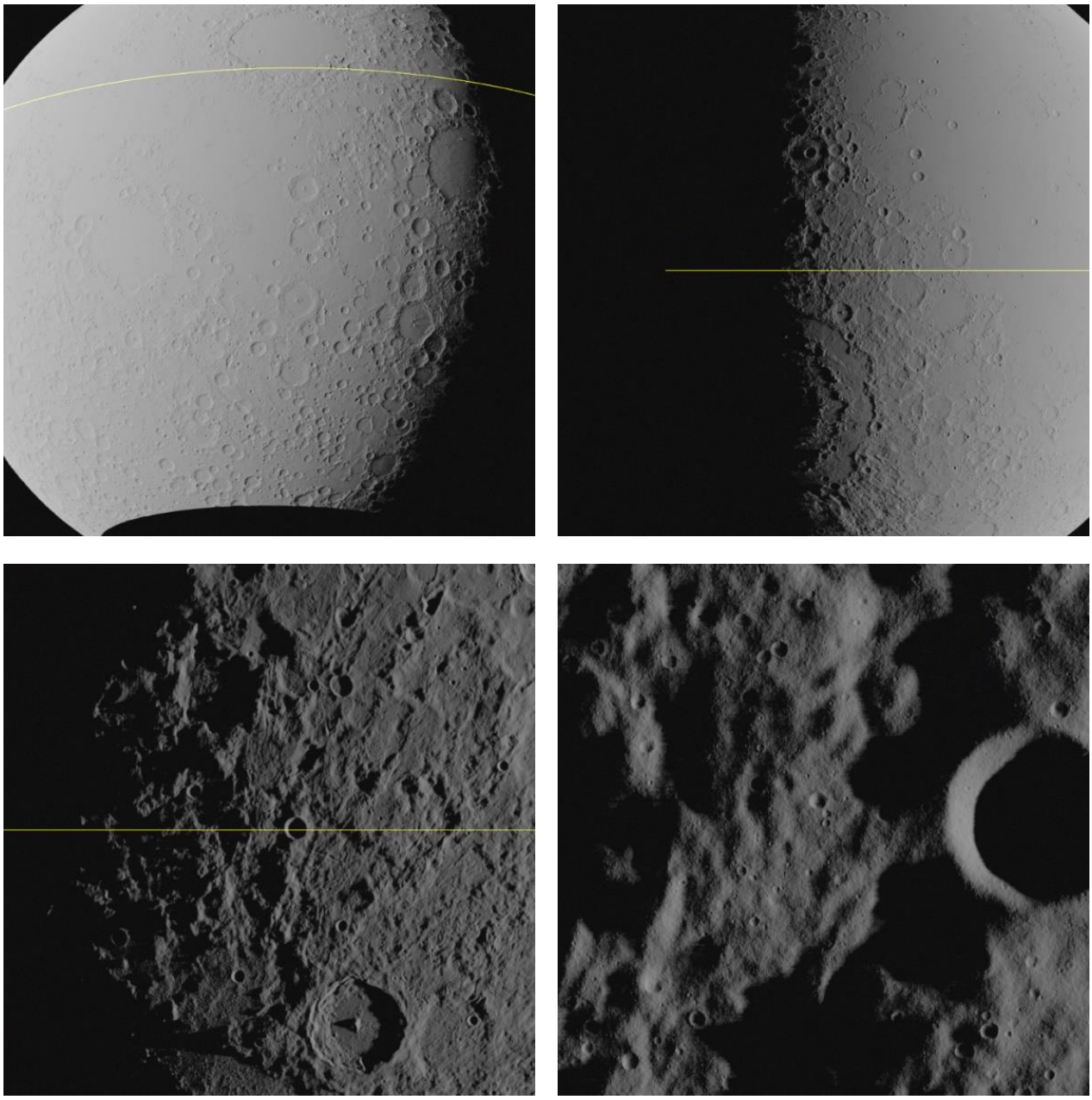


Figure 5: SLDEM at different distances showing the orbit flight path

2.4 IMPORTING MULTI-LOBED OBJ MODELS

PANGU v6 has existing functionality to import OBJ shape models into the ICQ (Implicitly Connected Quadrilateral) format. The ICQ format allows PANGU surface modelling enhancements to be applied because neighbouring vertices can efficiently be located (unlike OBJ format). However, this is limited to single-lobed objects which don't have ambiguity when assigning a spherical polar coordinate to points on the surface. Importing arbitrarily shaped genus 0 triangle meshes into ICQ format is a challenging task, particularly when they have multiple lobes such as comet 67P/Churyumov-Gerasimenko or asteroid 433 Eros. The conversion process is, in general, non-trivial and an area of active research. Different methods have been proposed for converting between an arbitrary mesh and a simplified mesh (such as a sphere or a plane) with different requirements on which properties must be preserved (*e.g.* angles or lengths).

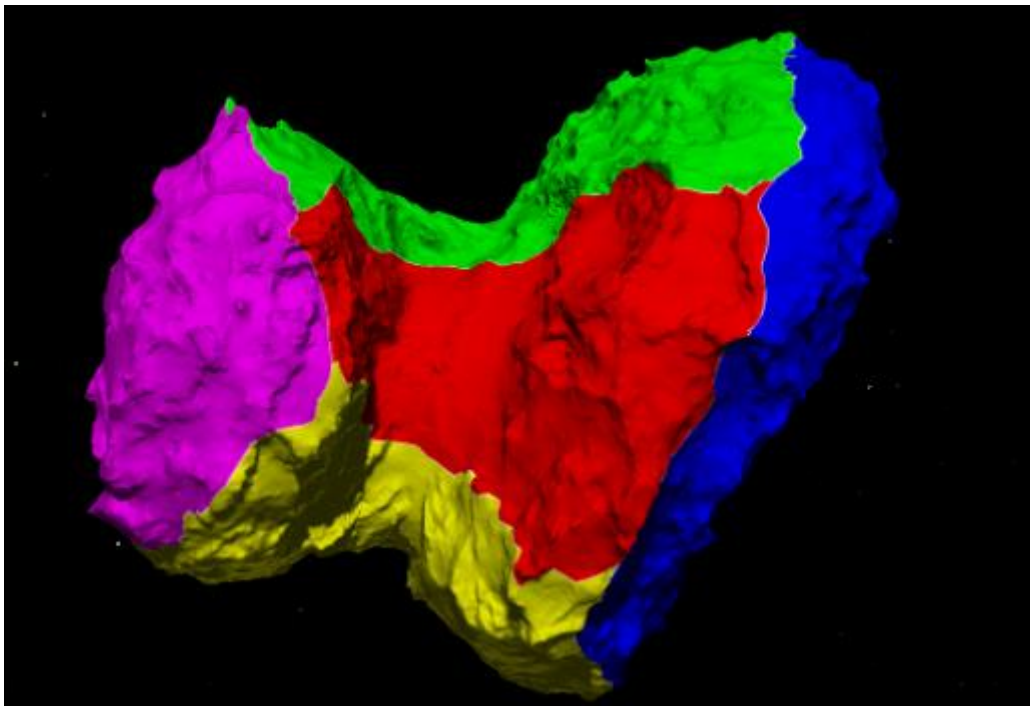


Figure 6: PANGU OBJ/ICQ prototype disk decomposition of comet 67P

The process chosen for this work is based on disk growth to separate the mesh into six pieces (one for each face of the ICQ), followed by 2D parameterisation, and then interpolation of the ICQ grid vertices into parameterised input mesh. The main difficulties of this approach are finding a robust way to assign 2D values to every vertex of a disk while maintaining consistency for edge vertices, and a robust way to identify the original mesh vertices for interpolation of the 3D position for a given ICQ point. Two main issues were identified with the prototype implementation. The first is that the simplification process does not yield a "nice" mesh and can result in poor or degenerate parameterisations; the second is that the interpolation process uses barycentric coordinates which

suffers from extrapolation problems, especially when the mesh has not been “nicely” simplified. This research implementation, implemented as a prototype mode of the PANGU viewer, shows that this approach can work well for detailed meshes of complex bodies such as 67P. In Figure 6 an OBJ mesh of 67P with 3.6 million triangles, has been imported and decomposed into separate ICQ faces. However, some meshes of the same body may fail to simplify or may produce unwanted artefacts in the generated ICQ model such as those highlighted in Figure 7.

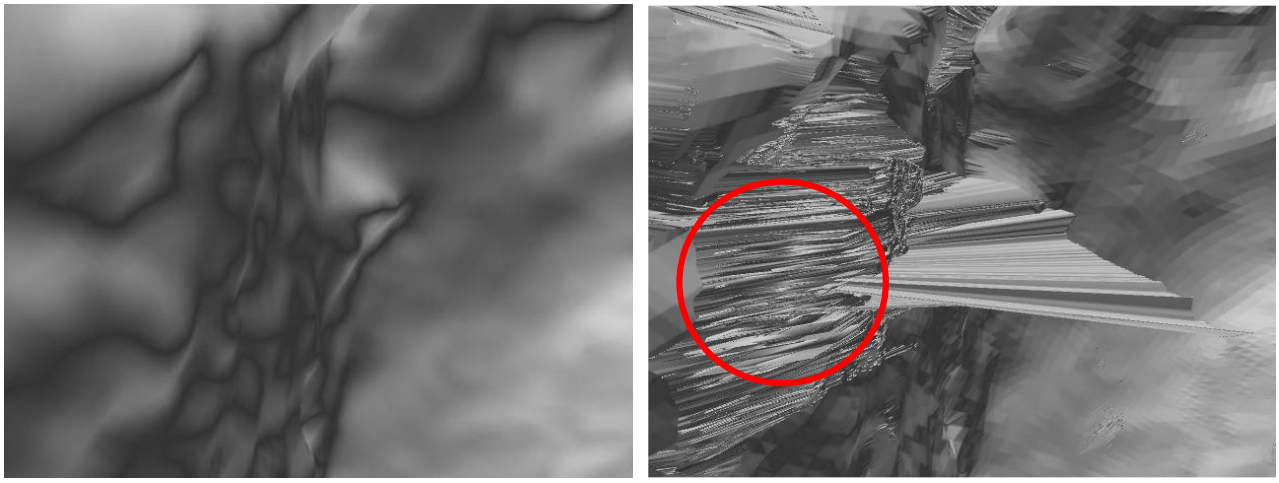


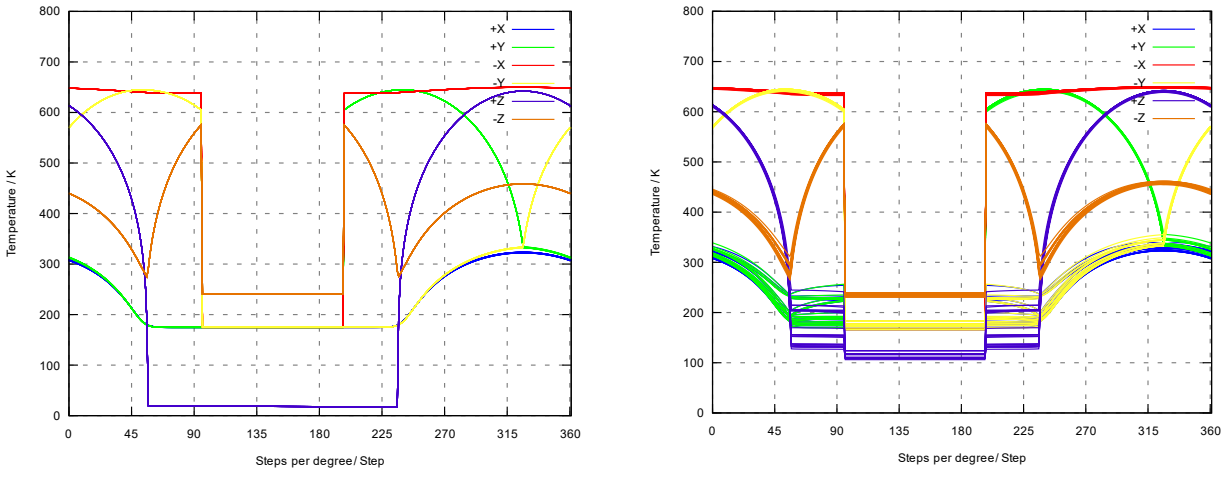
Figure 7: Original 67P mesh (left) and ICQ artefacts (right)

3 ENHANCED THERMAL SIMULATION OF SPACECRAFT SIMULATION

To obtain further data to improve the realism of thermal simulations of spacecraft in orbit, two thermal engineering software packages were reviewed to obtain thermal data from orbiting spacecraft simulations; SolidWorks and ESATAN.

SolidWorks is a powerful, feature-rich, engineering simulation tool which is capable of simulating complex CAD models. It contains the Flow Simulation thermal module which includes functionality to define different external heat sources, including solar. The SolidWorks experiments showed that SolidWorks could be useful for simulating material and thermal shadow effects to compare with, and validate, PANGU thermal simulations of spacecraft objects in limited scenarios. However, it was less appropriate for simulating realistic orbital scenarios because it was difficult to include anything other than a simplified Earth albedo effect that did not include the per-face Earth partial visibility factors.

ESATAN has limitations with model complexity size and sample resolution, but it does contain an orbital module which can simulate temperatures in orbital simulations of simple models. From this we were able to evaluate PANGU spacecraft thermal simulations. The ESATAN experiments confirmed that the PANGU v6 stateless transitions for MLI materials was realistically fast for objects going into, and out of, eclipse as shown by new experiments with a higher sampling rate than was previously able to be obtained.



MLI (no conduction mode)

MLI (conduction mode)

Figure 8: ESATAN CubeSat temperatures

However, using ESATAN's "conduction" mode, where limited conduction is simulated between mesh objects, we observed that the coldest face of a CubeSat in eclipse in an Earth orbit will have a substantially higher temperature than the "no-conduction" mode; the "no-conduction" mode closely matches PANGU v6 results where conduction between faces is not simulated. This is shown in

Figure 8 with the left graph showing the temperatures from the six faces of an Earth orbiting CubeSat in ESATAN in “no conduction” mode which from which PANGU v6 thermal spacecraft simulations have been validated. Each face contains a 5×5 grid of temperature sensors which are all plotted in the graph. The right graph shows the same simulation in ESATAN in conduction mode and shows warmer temperatures on the coldest face during eclipse, and a small spread of temperatures across faces where there are multiple values of a colour at a single sample position.

We surmise that the warmer temperature on the right graph for the coldest face during eclipse is due to limited conduction of Earth shine across faces at cold temperatures during eclipse. This conductive effect is not something that PANGU can currently simulate, but the effect could be approximated in future versions of PANGU. However, this is only likely to be important for simulating thermal cameras that can detect low temperatures *e.g.* those less than 100K. It is expected that thermal cameras used for vision-based navigation would not be able to detect temperatures less than about 250K. A small range of temperatures across CubeSat faces is also observed at some points in the simulations.

3.1 THERMAL SIMULATION EXTENSIONS

To enhance the thermal simulation of spacecraft, new features were implemented to add an optional additional scale factor that can be applied for indirect (reflected) albedo and planet thermal emission contributions based on object/planet distance, generalised texture nodes to allow separate VIS and TIR texturing, and support for separate VIS and TIR BRDFs.

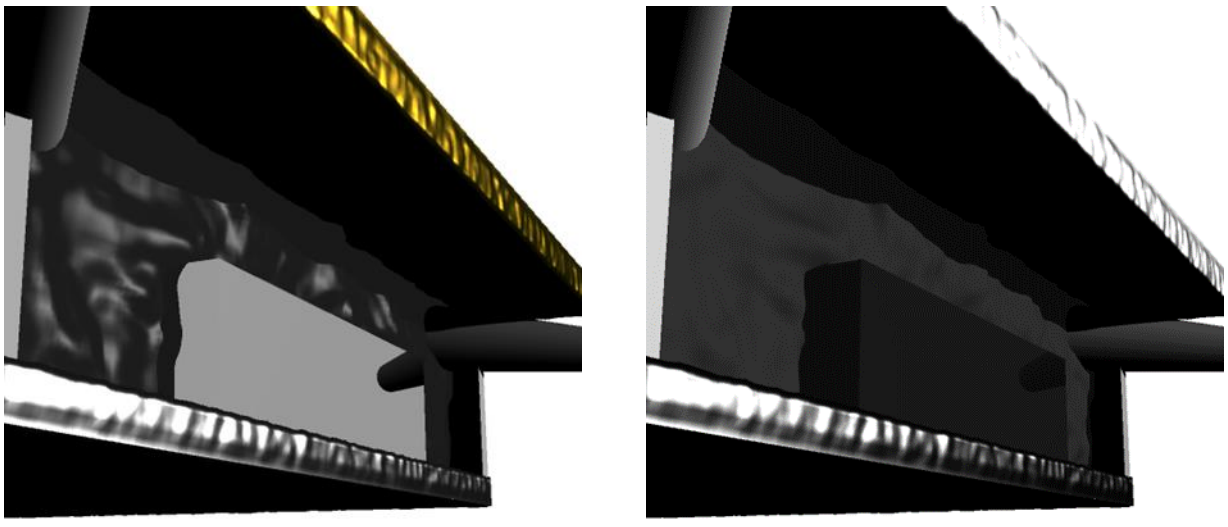


Figure 9: Separate VIS (left) and TIR (right) BRDFs on the same model

These generalised textures and separate VIS/TIR BRDF features enhance the realism of thermal simulations. Previously a model could be used for VIS simulations, or TIR simulations, but not both, otherwise it might appear unrealistic. The improvement is demonstrated in images of the PANGU example PRISMA/Tango model shown in Figure 9 which shows a VIS rendering of the model on the left and a TIR rendering of the same model on the right. The effects of the separate BRDFs can be observed in the VIS gold effect versus the TIR silver effect, the difference in the appearance of the black MLI surrounding the radiator, and the obvious difference in reflectivity of the radiator itself.

4 ENHANCED VERSATILITY ON CAMERA AND SENSOR MODELS

To enhance versatility on camera and sensor models, a new feature to add gamma correction to PANGU generated images has been implemented, and an “out-of-the-box”, sensor demonstration simulation example has been provided using MATLAB.

4.1 GAMMA CORRECTION

Currently PANGU rendered images are linear which is important for real images. But for display on-screen (*e.g.* physical for camera-in-loop testing), or for including in screenshots, PANGU ought to apply gamma correction. Without this, displayed images may be too dark or have the wrong brightness profile. It is important that images that are saved to disk or transmitted over TCP/IP are not gamma-corrected, in most cases, because they are intended to be processed by other systems. However, it could be useful for users to have the option to save or transmit gamma-corrected images.

Gamma correction must be the last operation that is applied to the image before it is saved or displayed; double correction must not occur. There are four use cases supported:

- normal GUI image display: apply DETECTOR gamma to detector model images
- save-to-disk: apply FILE gamma to detector model images
- transmit via TCP/IP: same as save-to-disk
- render-and-hold mode: apply HOLD gamma to displayed images not when first rendered

For convenience the gamma correction properties can be set separately for the different modes of operation: DETECTOR, FILE and HOLD. This allows, for example, images to be saved or transmitted over TCP/IP in linear mode while the render-and-hold display uses gamma correction.

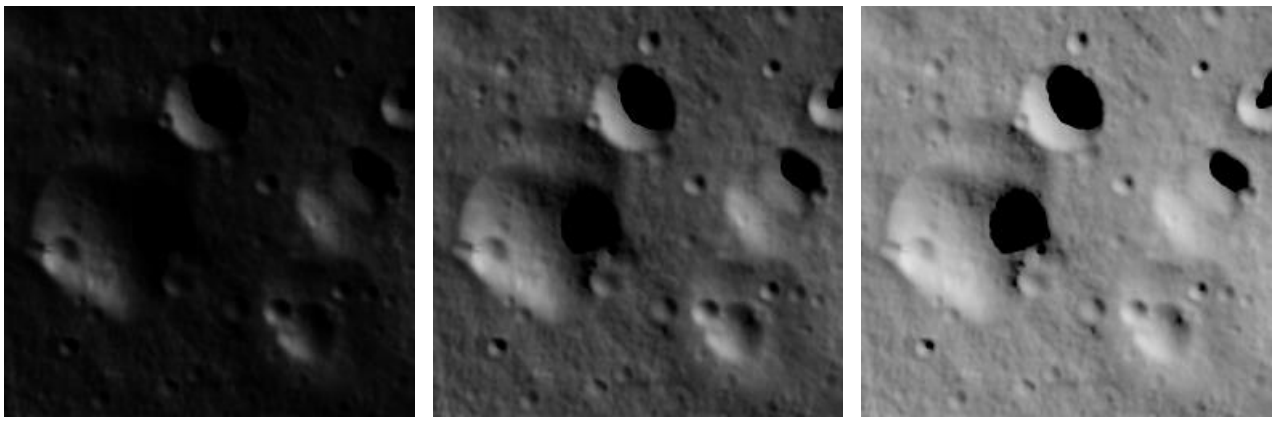


Figure 10: Gamma correction darkening (left), original (centre), and brightening (right)

Except for the special render-and-hold mode, only detector model images can be gamma corrected: simple radiance-only images will not use gamma correction. However, it is easy for the user to use

a pass-thru detector model configuration if gamma correction of radiance images is desired. Enabling gamma correction for radiance-only imaging would require rendering to a separate frame buffer which would increase system complexity, and reduce performance, for this mode. An example showing both lightening (left) and darkening (right) PANGU gamma correction is shown Figure 10, with the reference image shown in the centre.

4.2 SENSOR SIMULATION EXAMPLE

A case study example has been provided to run a dynamic sensor simulation in a MATLAB client connected to a PANGU server which generates the simulated images in real-time. It uses the existing **lunar_surface** model example as a suitably cratered terrain, with high-resolution regions and surface boulders included the model. A PANGU script is provided which loads and runs the **lunar_surface** simulation in server mode.

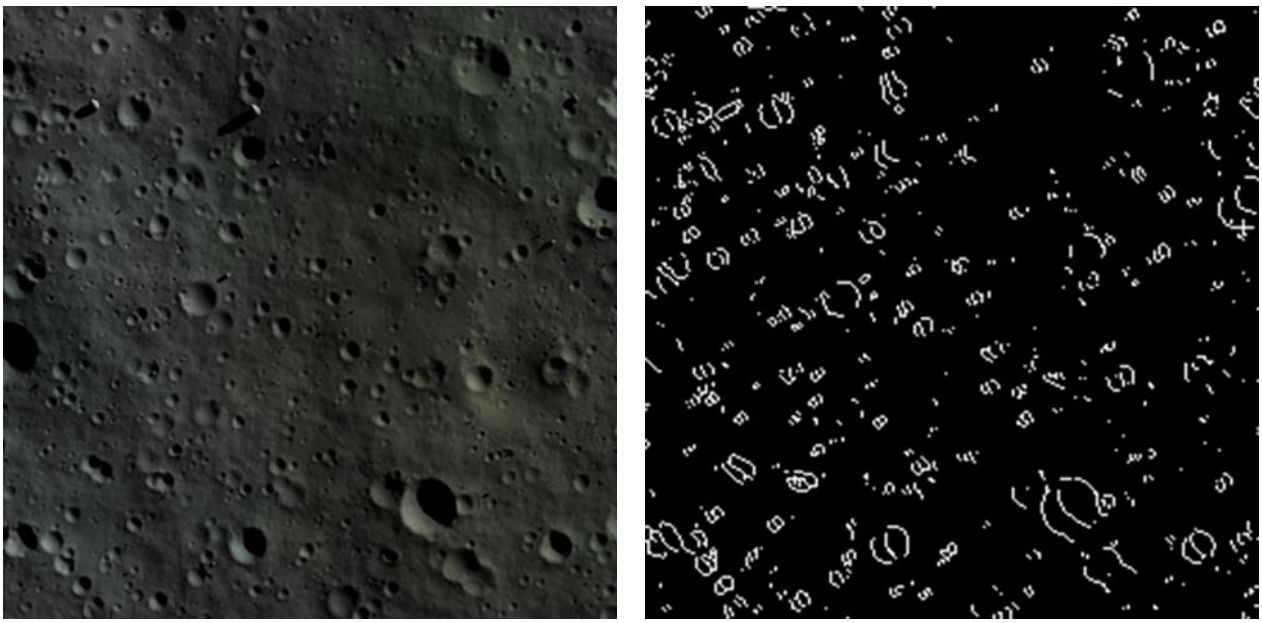


Figure 11: Sensor simulation MATLAB example

The MATLAB example connects to the PANGU server, obtains images through a TCP/IP socket connection, and then runs a Sobel edge detection function on the PANGU images as they are generated. This acts as an example of basic image processing in a real-time simulation. Figure 11 shows images displayed by the MATLAB sensor simulation example (within MATLAB) with the PANGU rendered image on the left and the MATLAB processed image on the right.

5 MAINTENANCE

Maintenance of PANGU was managed throughout the project with the NCR (Non-Conformance Report) system implemented prior to this project. PANGU users can report any issues though a website form, resulting in notifications to UoD and ESA. NCRs were recorded, discussed at NCR boards, prioritised, and corrected as required during the project. In addition, two significant maintenance enhancements were developed. An installation home directory placeholder symbol was added to simplify paths to PANGU directories in viewer settings, and the low-level rendering engine was updated from OpenGL 3.0 to OpenGL 3.2 to avoid using deprecated functionality.

5.1 VIEWER HOME DIRECTORY

To allow the viewer to find critical resources such as shaders, it has been updated to determine its installation directory on start-up. This directory is recorded in a special placeholder symbol `@{PANGU}` allowing the shaders folder to be specified as `@{PANGU}/shaders`. As part of the normal INI file checks, the viewer will attempt to process `@{PANGU}/pangu.ini` and `@{PANGU}/viewer.ini`. These files can provide important settings such as the shaders location; in practice only `pangu.ini` will exist; there is no need for a separate `viewer.ini`. Together, these three features ought to mean that users no longer need to explicitly specify the location of critical resources in their own INI files, making the viewer much more robust and easier to use especially for models at different directory levels.

5.2 UPDATE VIEWER OPENGL RENDERING TO GLSL #VERSION 150

PANGU v6 was designed to operate with GLSL #version 130 (OpenGL version 3.0) which was the last version that supported deprecated features such as the fixed-function vertex attributes. The first step is to allow PANGU to target GLSL #version 150 (OpenGL version 3.2) which eliminates fixed function rendering. This required changes to the PANGU GLSL shaders such as:

- texture1D(), texture2D(), shadow() etc are replaced by an overloaded texture() function
- vertex attributes such as gl_Vertex, gl_Normal have been replaced by user-defined attributes
- uniforms such as glLightModel were replaced by user-defined uniforms

This update does not give immediate benefits to PANGU, but this essential maintenance provides a solid foundation on which future updates can be made such as updating all instances where fixed-function rendering is used with a shader-based alternative. Once that has been achieved, the renderer will be able to operate purely in shader mode or fixed-function (no-shader) mode which should theoretically provide more opportunities for increased performance.

6 SUMMARY AND CONCLUSIONS

This activity has resulted in substantial improvements to PANGU through planned enhancements and maintenance activities to ensure that PANGU continues to be a useful tool for future spacecraft landing missions. The combination of new functionality for high-fidelity rendering with relevant case study examples, enhancements to the thermal rendering of spacecraft which was first developed for PANGU v6, and added support for sensor simulation and essential maintenance has ensured that PANGU can continue to be ideally placed to be used to verify and validate vision-based navigation systems throughout the design and qualification phases of ESA space programmes using real-time avionics test benches and other approaches. It can be used to support landing missions on the Moon, Mars and asteroids and spacecraft orbital simulations using both visual and thermal cameras.

7 DOCUMENT CHANGES

This is the first release of this document.