PANGU: VIRTUAL SPACECRAFT IMAGE GENERATION

Nick Rowell[1], **Steve Parkes**[1], **Martin Dunstan**[1], **and Olivier Dubois-Matra**[2]

[1]*Space Technology Centre, University of Dundee, Dundee, UK*
[2]*ESA (ESTEC), 2200 AG Noordwijk, The Netherlands*

ABSTRACT

The Planet and Asteroid Natural scene Generation Utility (PANGU) is a set of software tools that provide a complete modelling and sensor simulation environment to support the development and testing of vision based guidance systems. It includes a suite of 3D modelling functions for producing realistic planetary surfaces and asteroids, complete with morphological features such as craters and boulders. It also provides a viewer tool that can be used to generate synthetic camera, RADAR and LIDAR measurements from any viewpoint, allowing PANGU to be used in closed loop engineering simulations of complete GNC systems.

Recently, the modelling and rendering functions of PANGU have been enhanced to include artificial materials and complete spacecraft models. This includes the ability to import spacecraft models from CAD files, render highly non-Lambertian surfaces such as Solar panels and OSR tiles, and include articulated joints and independently moving parts that can be configured dynamically. These additions are intended for use in scenarios such as formation flying and in-orbit rendezvous and docking. In this paper, we review the recent enhancements to the PANGU toolset and their role in producing high quality simulated images of spacecraft.

Key words: PANGU; Virtual Spacecraft; Optical Navigation.

1. INTRODUCTION

The use of computer simulation to provide synthetic sensor data for testing guidance systems offers several advantages over more traditional techniques, such as scale models. For example, ground truth sensor motion is known to arbitrary precision, whereas even the best scale models are subject to engineering tolerances and suffer some degree of flexure. The ability to regenerate artificial terrain with ease allows computer simulations to cover a wider range of scenarios, and simulated sensor measurements are essentially noise-free, allowing users to employ their own tightly controlled noise models and tune guidance algorithms for optimal performance.

In this spirit, PANGU was developed originally to investigate the feasibility of simulating computer images of the Lunar surface for testing vision-based lander navigation. Since then, PANGU has been extensively developed over several European Space Agency (ESA) funded projects[1] to support the simulation of asteroids, whole planets, RADAR and LIDAR sensors, fog and dust extinction, and shadows. There are two main areas of research involved in producing realistic imagery of simulated planetary surfaces: first, techniques for generating artificial terrain models, complete with craters, boulders and other types of morphological feature, and second, techniques for simulating realistic sensor measurements under a wide range of viewing conditions.

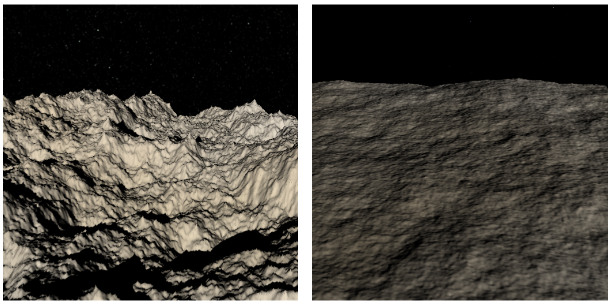
1.1. Modelling planetary surfaces

The starting point for building realistic planetary surfaces is a model of the underlying terrain elevation. This is generated by distorting an initially smooth grid by the addition of random height offsets, in order to produce structure similar to that seen in nature. It is widely accepted that fractals capture the self-similarity evident in natural phenomena (Mandelbrot 1983), and PANGU uses several variations on fractional Brownian motion to produce realistic terrain elevation models. A single parameter controls the degree of roughness of the terrain and the amount of structure at all scales. Figure 1 shows two such terrain models.

Various types of morphological feature are then added to the underlying terrain. Many common types of geological feature are modelled by PANGU, including craters, boulders, sand dunes and lava flooding. These can be placed individually by the user, or automatically according to distribution functions taken from planetary science literature, so that the statistical properties of the model match those seen in nature as closely as possible.

For example, for simple craters PANGU adopts the shape models of Melosh (1996) and McGetchin et al. (1973), and includes the crater rim, bowl and ejecta blanket. The

[1] LunarSim study (ESA Contract No. 12821/98/NL/MV); PANGU study (ESA Contract No. 11747/95/NL/JG); Asteroid and Whole Planet Simulation with PANGU study (ESA Contract No. 17338/03/NL/LvH/bj).

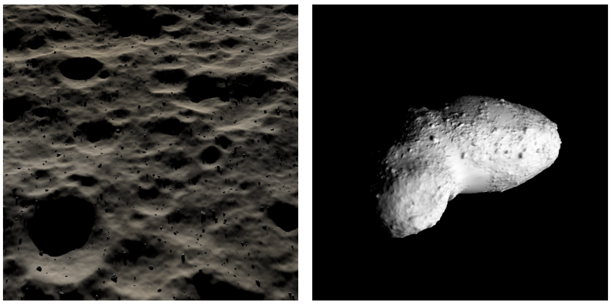


(a) Rough

(b) Smooth

Figure 1. PANGU uses fractal terrain generation techniques to make realistic surface elevation models.

model is parameterised in terms of the crater radius and age, with older craters degraded due to gradual infill by ejecta from younger craters (e.g. Ross 1968; Soderblom 1970). The crater shape is further modified by the shape of the underlying terrain, and multiple craters are added in reverse order of age so that younger craters obliterate older craters. Figure 2a shows a PANGU surface model with craters distributed in size according to the results of Hartmann (1995), appropriate for the Lunar Mare regions.



(a) Lunar type surface

(b) Itokawa shape model

Figure 2. Synthetic Lunar-type surface generated by PANGU (2a) with craters and boulders; shape model of asteroid 25143 Itokawa imported for image generation (2b).

PANGU can also import existing terrain elevation models, e.g. MOLA or LRO digital elevation models for the surface of Mars and the Moon. These can be processed to add high-resolution detail or remove artefacts, producing models more suited to image generation. In addition, PANGU can produce and/or import asteroid models (see Figure 2b).

1.2. Sensor simulation

PANGU is capable of simulating camera, LIDAR and RADAR instruments, as well as performing back-projection and line of sight queries to provide ground truth for e.g. feature tracking or 3D reconstruction algorithms. The LIDAR and RADAR models are derived from common types of sensor, and the interested reader is referred to Parkes et al. (2003) and Parkes et al. (2005) for more details.

PANGU models are based ultimately on a triangulated polygonal mesh, and image generation uses polygon shading techniques and texture mapping to produce raster images (see Figure 3). The Open Graphics Library (e.g. Shreiner et al. 2008, "The Red Book") is used to speed up the rendering process by utilising the hardware acceleration capabilities of modern graphics cards. Image resolution, in terms of the sensor size and field of view, are fully configureable by the user, though the dynamic range of each pixel is limited to 24 bits (8 in each of the RGB channels). There is also no concept of exposure time: users simply set the relative brightness of light sources, and the resulting pixel levels are essentially a measure of the intensity of reflected light. The projection model is that of a pinhole camera, resulting in images free from radial and higher order distortions, although perspective distortion can be significant for short focal lengths or wide field-of-view images. Images are also free from optical defects, pixel noise and depth-of-focus effects (everything is in focus regardless of distance).

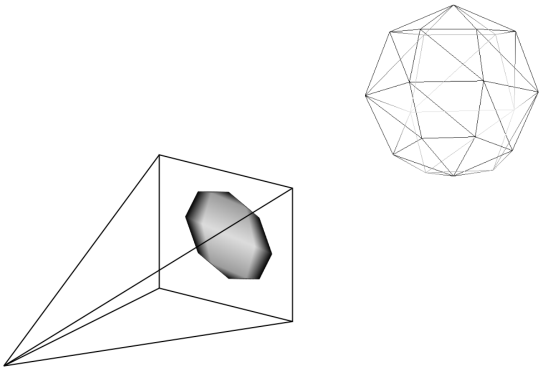


Figure 3. Image depicting rendering of a polygonal mesh-based object to a shaded raster image. Perspective projection is used, producing rectilinear images with no high order optical distortions, depth-of-focus or pixel noise.

The image generation tool is also capable of simulating atmospheric effects such as fog and dust extinction.

1.3. Closed loop testing

The sensor simulation capabilities of PANGU can be included in closed loop engineering simulators. In such a scenario, the trajectory of the camera is not known in advance and images must be generated on-the-fly as the camera moves in response to control signals and the environment.

To support this, PANGU can run in server mode. Clients such as engineering simulators connect over a TCP/IP interface to perform such operations as setting the pose

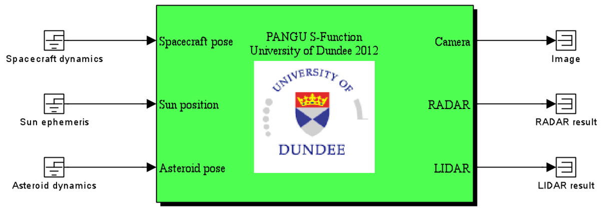


Figure 4. PANGU has been incorporated in Simulink models for use in GNC simulators via an S-Function.

of the camera and target bodies, and requesting images and other measurements. A Java client library is provided with PANGU, which allows easy integration with e.g. Matlab and Simulink. For example, in Simulink models the necessary Java client messages can be placed in an S-Function wrapper, providing a simulation block that accepts the camera position and orientation as input, and outputs image, RADAR and LIDAR measurements—see Figure 4. PANGU has been used in a variety of ESA projects in this manner.[2]

2. VIRTUAL SPACECRAFT IMAGE GENERATION

At present there is considerable interest in developing reliable rendezvous and docking systems for spacecraft, for use in, for example, in-orbit servicing of satellites (e.g. Leinz et al. 2008), autonomous capture of sample return canisters and precision formation flying. Machine vision techniques provide an attractive solution, as they potentially supply accurate pose estimation and tracking for relative navigation of two or more spacecraft (e.g. Alonso et al. 2000; Kelsey et al. 2006). In order to support the testing of these scenarios, several broad extensions to the PANGU toolset have been developed.[3] These include the ability to build PANGU models of spacecraft in a straightforward manner. Several enhancements aimed at improving image generation have been incorporated, such as the ability to simulate the appearance of a variety of construction materials. In this section, we introduce the new features of PANGU and describe their use in producing realistic simulated images of spacecraft.

2.1. Material reflectance functions

In computer graphics, different materials are simulated by altering the way that light reflects from their surface. Reflectance functions are parameterised in terms of the directions of incident and reflected light, and a variety of such Bidirectional Reflectance Distribution Functions (BRDFs) have been published over the years designed to simulate different types of material. These are split roughly into two broad categories representing diffuse and specular reflectance contributions, which are combined in different ratios to produce different types of effect. In the past, PANGU has supported the simple isotropic Lambert BRDF for diffuse reflectance, and the Blinn-Phong BRDF for specular reflectance. In addition, the diffuse Hapke BRDF (Hapke 1981) was added to allow more accurate shading of Lunar regolith, where the "opposition effect" introduces a strong backscattering component. These BRDFs provide a good set of shading models for Solar system planetary surfaces, but present significant limitations when artificial materials such as metallic surfaces, cloth MLI and Solar panels are to be simulated.

PANGU has been upgraded to include a range of extra BRDFs designed to simulate a variety of materials used in the construction of spacecraft. Figure 5 demonstrates the new Cook-Torrance specular BRDF (Cook & Torrance 1981) used to simulate surfaces with microfacets. This has a range of tuneable parameters that allow the user to manually configure the shading to achieve the desired effect.

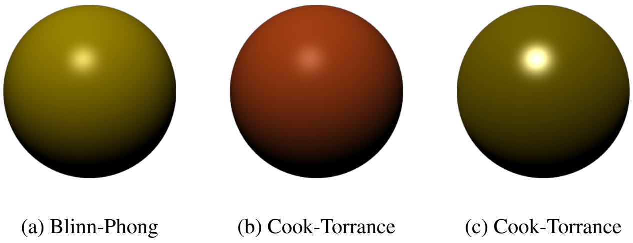

(a) Blinn-Phong
(b) Cook-Torrance
(c) Cook-Torrance

Figure 5. Sphere rendered with original Blinn-Phong specular highlight (5a) and new Cook-Torrance function to simulate dull plastic material (5b) and shiny metallic surfaces (5c).

Figure 6 demonstrates the new Oren-Nayar diffuse BRDF (Oren & Nayar 1994) used to simulate rough surfaces such as cloth or plaster, as well as the existing Lambert and Hapke BRDFs.

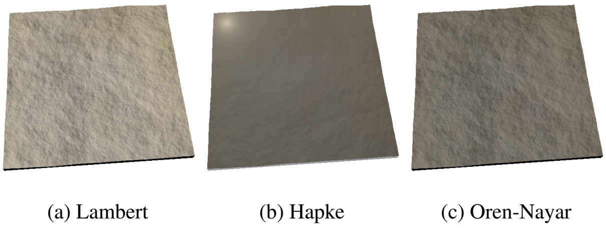

(a) Lambert
(b) Hapke
(c) Oren-Nayar

Figure 6. Surface rendered with diffuse BRDFs: standard Lambert model (6a); Hapke model with strong backscattering component (6b); Oren-Nayar model designed for rough surfaces (6c) such as cloth.

[2] e.g. NEOGNC study (ESA contract no. 23048/09/NL/AF)

[3] In the context of ESA contract no. RFQ/3-12351/08/NL/MP, "Virtual Spacecraft Image Generator Tool".

By combining diffuse and specular reflectance functions, a much richer range of effects can be achieved. For example, Figure 7 uses a combination of the Oren-Nayar and Cook-Torrance functions to simulate shiny foil material.

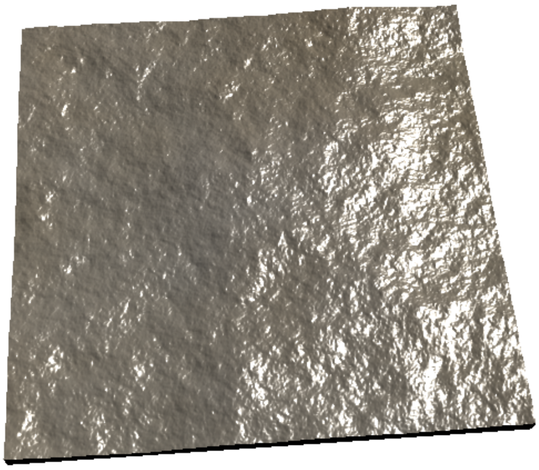


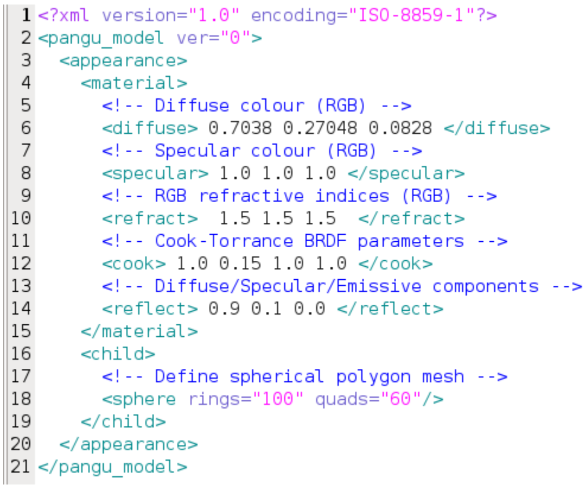
Figure 7. Example of combining diffuse and specular reflectance functions: the Oren-Nayar diffuse BRDF with Cook-Torrance specular highlights can simulate shiny foil material.

2.2. Building virtual spacecraft

Building realistic PANGU models of spacecraft is a very different problem to that of modelling natural terrain like planetary surfaces, for which stochastic methods based on fractals work well.

Two methods have been developed to allow users to build spacecraft models in PANGU. The existing PANGU tool `vrmltopan` allows users to convert fully triangulated VRML models to the `pan` format used internally by PANGU. This is designed to support users who have existing CAD models of spacecraft and who wish to import these directly to PANGU for rendering.

Alternatively, an `xml` based modelling language has been developed for the latest version of PANGU that allows users to define geometric entities in a natural way. These can range in complexity from single vertices and triangular strips, to various standard polygonal mesh objects and entire existing PANGU models. The material properties are added via `xml` tags and can be defined separately for all distinct components of a model. Components can be scaled, rotated and translated so that multiple instances of the same shape can be reused and have material properties set globally. These `xml` directives are saved in a text file with a `.pxn` extension, and a new tool called `pxntopan` converts this into a `pan` file for rendering in PANGU.



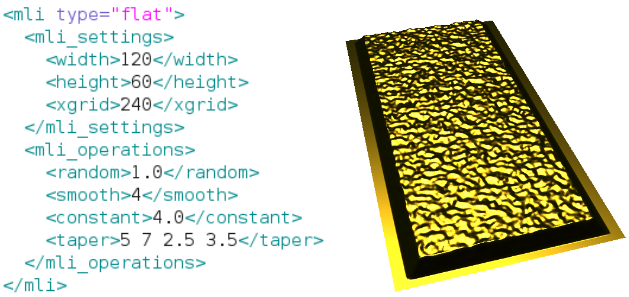
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<pangu_model ver="0">
  <appearance>
    <material>
      <!-- Diffuse colour (RGB) -->
      <diffuse> 0.7038 0.27048 0.0828 </diffuse>
      <!-- Specular colour (RGB) -->
      <specular> 1.0 1.0 1.0 </specular>
      <!-- RGB refractive indices (RGB) -->
      <refract>  1.5 1.5 1.5  </refract>
      <!-- Cook-Torrance BRDF parameters -->
      <cook> 1.0 0.15 1.0 1.0 </cook>
      <!-- Diffuse/Specular/Emissive components -->
      <reflect> 0.9 0.1 0.0 </reflect>
    </material>
    <child>
      <!-- Define spherical polygon mesh -->
      <sphere rings="100" quads="60"/>
    </child>
  </appearance>
</pangu_model>
```

Figure 8. XML definition of PANGU model in Figure 5b.

Figure 8 shows the `pxn` file that was used to generate the PANGU model in Figure 5b. There are separate `xml` tags for defining the diffuse and specular colours, as well as setting parameters for the various BRDFs that are available. The single geometric entity in this case is a spherical polygonal mesh.

2.3. MLI modelling

There is a subtle problem in rendering realistic images of spacecraft based on CAD models. Many sections of a spacecraft are wrapped in Multi-Layer Insulation (MLI) during the final construction stages. MLI generally has a rough, wrinkled or embossed outer surface and the detailed shape of this is not included in the CAD model, which will simply show flat panels. We have developed a technique to simulate the appearance of sections of MLI, using an approach involving randomized wrinkling and embossing of an initially flat sheet. The processed sheet is then formed into one of a set of basic shapes, such as conical or tubular lengths, and finally imported into a `pxn` file and wrapped onto an existing spacecraft component by carefully defining the position and orientation. This is all controlled via a dedicated `xml` tag.



```
<mli type="flat">
  <mli_settings>
    <width>120</width>
    <height>60</height>
    <xgrid>240</xgrid>
  </mli_settings>
  <mli_operations>
    <random>1.0</random>
    <smooth>4</smooth>
    <constant>4.0</constant>
    <taper>5 7 2.5 3.5</taper>
  </mli_operations>
</mli>
```

Figure 9. Modelling a flat sheet of MLI with tapered edges.

An example of a flat sheet of MLI with tapered and flattened edges is shown in Figure 9, along with the `xml` tags used to define it. The `<mli type="flat">` tag declares a new section of MLI and sets the overall shape. The `<mli_settings>` tag controls the size and resolution of the initial sheet, and the `<mli_operations>` tag encloses the various wrinkling, smoothing and tapering operations that are applied to the initial sheet to simulate the MLI appearance. In this figure, the resulting MLI sheet is rendered with the Cook-Torrance BRDF to add the shiny metallic finish.

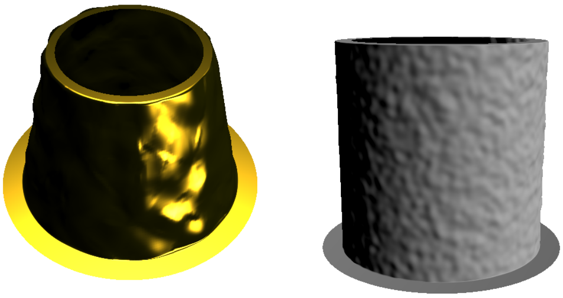


Figure 10. Modelling conical and tubular sections of gold and cloth MLI with flanged ends.

2.4. Long range views

One of the main difficulties in rendering mesh-based objects is matching the resolution of the mesh to that of the pixels in the final image. No polygon in the mesh should be less than one pixel in size otherwise aliasing may occur. In the context of rasterising systems like OpenGL (as used by PANGU), only one polygon of the mesh can determine the colour of an individual pixel. If more than one polygon falls inside the same pixel, then one of the polygons will be effectively selected at random to determine the pixel colour (see Figure 11). The result is that distant spacecraft models will flicker as they move.

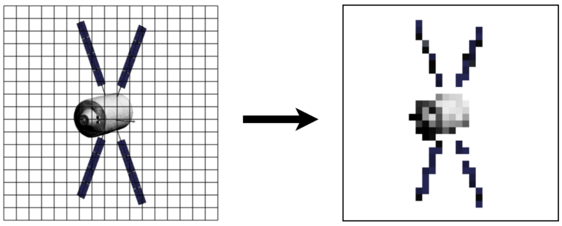


Figure 11. For distant views of spacecraft, the target may subtend only a few pixels (left). When many polygons fall in the same pixel, the rendering system selects one of them at random to determine the colour of the entire pixel. The final image exhibits significant aliasing (right) and loss of colour resolution.

To get around this, PANGU now provides the option of using intermediate 'imposter' images when rendering spacecraft (and any other type of PANGU model). An image of the spacecraft is taken from the same camera position as the original, but swivelled and zoomed in to produce a much higher resolution image where the spacecraft fills the frame. This 'imposter' image is drawn onto a rectangular polygon quad, and this quad is then rendered into the original image at an appropriate size—see Figure 12. The larger polygons of the imposter quad cover multiple pixels, so the problems of aliasing and loss of colour resolution are avoided. Also, when drawing textures into a frame, OpenGL interpolates colours between pixels, providing an additional stage of filtering and resulting in very smooth images.

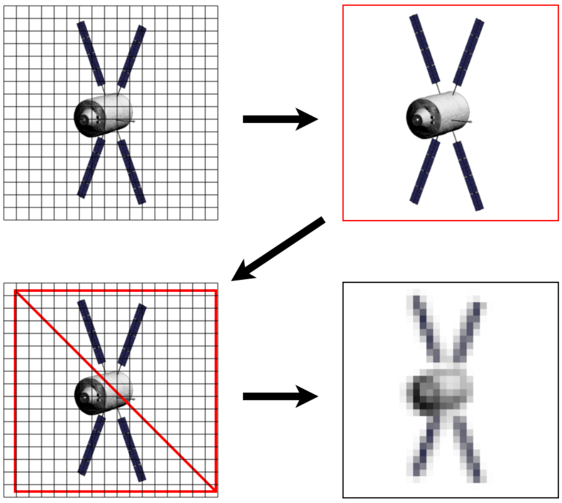


Figure 12. When rendering long range views of spacecraft, an intermediate image of the spacecraft is first rendered at high resolution (top right). This 'imposter' image is then drawn onto a large polygon quad (red triangles, bottom left) and this is painted into the final image (bottom right). By using imposter images, long range views of spacecraft can be rendered free from aliasing.

In implementing this scheme, care has been taken to ensure that imposters are correctly handled when multiple targets overlap, and when targets include mirrored and/or shadowed surfaces.

3. DISCUSSION

3.1. Considerations for GPU based rendering

Graphics cards are generally intended for rendering terrestrial scenes, where the ratio of the distance to the nearest and farthest objects is small, and transformation matrices can be stored in hardware as single- or half-precision floating point numbers with sufficient accuracy.

When rendering space scenes, the nearest object may be metres in front of the camera and the most distant thousands of kilometres away. If not carefully handled, such scales can lead to significant artefacts due to numerical errors, such as quantized positions of objects and incorrect shadows. PANGU regularly works in regimes well outside the normal range of standard GPUs, and has been carefully designed to avoid these problems.

3.2. How to access PANGU

PANGU licenses are available from ESA for use on ESA projects. Standalone licenses can also be purchased from STAR-Dundee.[4]

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the European Space Agency for the work described in this paper, in particular ESA project managers Christian Philippe, Olivier Dubois-Matra, Salvatore Mancuso and Stein Strandmoe. This work was carried out under ESA contract no. RFQ/3-12351/08/NL/MP, “Virtual Spacecraft Image Generator Tool”.

REFERENCES

Alonso, R., Crassidis, J. L., & Junkins, J. L. 2000, in American Institute of Aeronautics and Astronautics GNC Conference & Exhibit, Paper No. AIAA 2000-4439, Denver, Co.

Cook, R. L. & Torrance, K. E. 1981, in Proceedings of the 8th annual conference on Computer graphics and interactive techniques, SIGGRAPH ’81 (New York, NY, USA: ACM), 307–316

Hapke, B. 1981, J. Geophys. Res., 86, 3039

Hartmann, W. K. 1995, Meteoritics, 30, 451

Kelsey, J., Byrne, J., Cosgrove, M., Seereeram, S., & Mehra, R. 2006, in Proceedings of the 2006 IEEE Aerospace Conference

Leinz, M. R., Chen, C., Beaven, M. W., et al. 2008, in Sensors and Systems for Space Applications II, Proceedings of the SPIE, Vol. 6958

Mandelbrot, B. 1983, The Fractal Geometry of Nature (New York: W. H. Freedman and Co.)

McGetchin, T. R., Settle, M., & Head, J. W. 1973, Earth and Planetary Science Letters, 20, 226

Melosh, H. J. 1996, Impact Cratering: A Geologic Process (Oxford University Press)

Oren, M. & Nayar, S. K. 1994, in Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH ’94 (New York, NY, USA: ACM), 239–246

Parkes, S. M., Dunstan, M., & Martin, I. 2005, in Proceedings of the Data Systems in Aerospace Conference (DASIA), Edinburgh, Scotland

Parkes, S. M., Dunstan, M., Matthews, D., Martin, I., & Silva, V. 2003, in Proceedings of the Data Systems in Aerospace Conference (DASIA), Prague, Czech Republic

Ross, H. P. 1968, Journal of Geophysical Research, 73, 1343

Shreiner, D., Woo, M., Neider, J., Davis, T., & the OpenGL Architecture Review Board. 2008, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2.1 (Addison-Wesley)

Soderblom, L. A. 1970, Journal of Geophysical Research, 75, 2655

[4] `http://www.star-dundee.com/contact-us`