NASA/TM-20230016076

Lunar Terrain Mapping Using 3D Software and Modeling Techniques for Glenn Research Center Communication Analysis Suite

Ana Granite and Bryan W. Welch
Glenn Research Center, Cleveland, Ohio

February 2024

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., "quick-release" reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 757-864-6500

- Telephone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Program
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM-20230016076

Lunar Terrain Mapping Using 3D Software and Modeling Techniques for Glenn Research Center Communication Analysis Suite

Ana Granite and Bryan W. Welch
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

February 2024

Acknowledgments

Ana Granite acknowledges NASA and the Office of STEM Engagement (OSTEM) for funding her research and 3D modeling experience at NASA's Glenn Research Center. The authors acknowledge the joint efforts of other team members, Chloe Imbusch, Sam Bloomingdale, and Mackenzie Martin, in working towards visualization enhancement of the GCAS.

Level of Review: This material has been technically reviewed by technical management.

Lunar Terrain Mapping Using 3D Software and Modeling Techniques for the Glenn Research Center Communication Analysis Suite

Ana Granite[*] and Bryan W. Welch
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

As NASA prepares to return humans to the Moon as part of upcoming Artemis missions, engineers must examine the surface features and safety of lunar terrain at various locations. The Glenn Research Center Communication Analysis Suite (GCAS) includes visualization tools developed to aid scientists in this quest to comprehend lunar terrain data, as well as spatial communication information.

Visual information and understanding will be highly useful to engineers as they use the GCAS to compare the topography of regions of interest for the Artemis III and future space travel missions. Not only can three-dimensional (3D) visualization display the intricacies of the lunar terrain, but it can also demonstrate relationships such as the presence or absence of communication links between Earth ground stations, lunar crew members, and rovers, and the presence of shadows on the lunar surface. In the future, the GCAS has the potential to support the planning of other space exploration missions as well. The emerging possibility of incorporating more planetary bodies becomes especially important as NASA plans not only to return to the Moon but to press onward to Mars.

Nomenclature

2D	two-dimensional
3D	three-dimensional
CGI	computer-generated imagery
DEM	Digital Elevation Model
GCAS	Glenn Research Center Communication Analysis Suite
LOD	Level of Detail
three.js	open-source JavaScript library used for displaying 3D graphics
UV	U-axis and V-axis

Introduction

The three-dimensional (3D) visualization capability in the Glenn Research Center Communication Analysis Suite (GCAS) supports using input data to create an interactive 3D interface that displays a third-person perspective of important components related to space exploration, such as the Moon, Earth, the Sun, Mars, satellites, and ground stations. The visualization software uses JavaScript® code (Oracle America, Inc.) and three.js, an open-source JavaScript library used for displaying 3D graphics. It is designed to communicate data on position, time, space, lighting, and other complex numeric and spatial

---

[*] NASA OSTEM Spring 2023 intern, Bowling Green State University undergraduate.

information about relationships between the lunar terrain and manmade assets, which can be computed automatically and understood quickly in a visual manner.

A previous iteration of the visualization capability was limited by its inability to illustrate the lunar surface as anything other than a spherical body. Therefore, the task at hand was to communicate lunar terrain and spatial data to both professional engineers and novice viewers with high accuracy, using software that would run efficiently and an interface that was easily understood and customizable to the user's needs. By interacting with the GCAS, the user could understand the relationships between objects and when and where communication links such as signals could be received. In addition to the lack of lunar terrain data, the earlier visualization tool also had the problems of a pinching texture and data losses at the poles due to a wedged U-axis and V-axis (UV) map and incompatible Mercator projection-type input, as well as fundamental limitations of the source data. Both problem areas needed refinements through generating rough terrain and 3D mapping to create a cohesive model that could be implemented for a third-person perspective of the Moon's surface in the GCAS.

3D Modeling Summary

The first avenue pursued for lunar terrain mapping was the use of 3D modeling and visual scripting in Blender®, an open-source 3D visualization development tool (Stichting Blender Foundation). The Shader and Geometry Nodes editors in Blender made it possible to script functions for the various specific input-output needs of 3D meshes. This process was simplified through the ability to experiment with the Python™ Scripting editor and to examine Blender's GitHub script to observe the inner mechanisms and functions within Blender (Ref. 1). The current Blender manual and earlier iterations also proved useful in navigating modes of using 3D computer-generated imagery (CGI) production software, which is often used in artistic settings, for a scientific purpose as well (Ref. 2).

The process of creating an accurate, complete 3D model of the Moon's surface using various types of data sets took plenty of trial and error and significant research. During development, bugs and limitations within the software were encountered, which created challenges that demanded alternate solutions. Individual aspects of the model were simultaneously tended to, such as displaying global data properly, addressing issues at the poles, projecting polar stereographic maps onto the globe, creating circular meshes from Digital Elevation Model (DEM) image inputs, aligning meshes, and decimating models, i.e., reducing the polygon count, to allow them to be used in a browser application. At the same time, we considered how decisions would affect outlying components of the model and how the model would be implemented within the software.

Export tests of crude model versions were conducted early on so that colleagues could test importing models and carry out Level of Detail (LOD) adaptive resolution display development. A .glb (GL Transmission Format Binary file) formatted file can contain information about 3D models, scenes, models, lighting, materials, node hierarchy, and animations. The discovery of incompatible methods of modeling for .glb file format export forced a restart halfway through the process, changing from visual scripted-based displacement to a different approach using Blender's modifiers.

Global Data Incorporation

Finding a way to address the pinched-looking texture that appeared at the poles (similar to the pinches shown in Figure 1) was a primary goal necessary to generate an accurate 3D model of the Moon. The first method attempted to achieve this goal involved editing the tiling of the UV map that dictates how a two-dimensional (2D) image input corresponds to a 3D mesh output.

Upon creation of a new UV Sphere to encompass all data in the source image, rectangular tile shapes were generated in the UV map to replace the previous mapping, which exhibited triangular tiles along the top and bottom of the map. Using the UV Editor, the map was adjusted to snap to the proper global data image ratio; the result is shown in Figure 2. Consequently, the UV map was unwrapped and rewrapped in such a manner that the projected image could correspond to the correct location on the sphere and account for compressed data at the poles. Next, using the Shader Editor allowed for implementing input texture and displacement data images found on NASA’s CGI Moon Kit visualization webpage (Ref. 3). Although this solution sufficed initially, flaws in this process had to be corrected later as the concept of creating a perfect global mesh from a Mercator-type image alone came into question. A close examination of the mesh at the poles revealed that warping persisted on a smaller level, as shown in Figure 3.

An intermediate attempt at eliminating the pinching observed at the poles involved bending a plane with a 2:1 width-to-height ratio twice horizontally and vertically into the shape of a sphere. However, imperfections at the poles continued, and new, flat circles appeared on opposite sides of the sphere, which are visible in Figure 4. These may have been issues involving altered normals, which represent the orientation of a given face, considering the impact that normals have on the simulation of light and the subsequent way light interacted with the mesh differently at these areas on the surface. Consequently, the base geometry produced was not a perfect spherical solution.

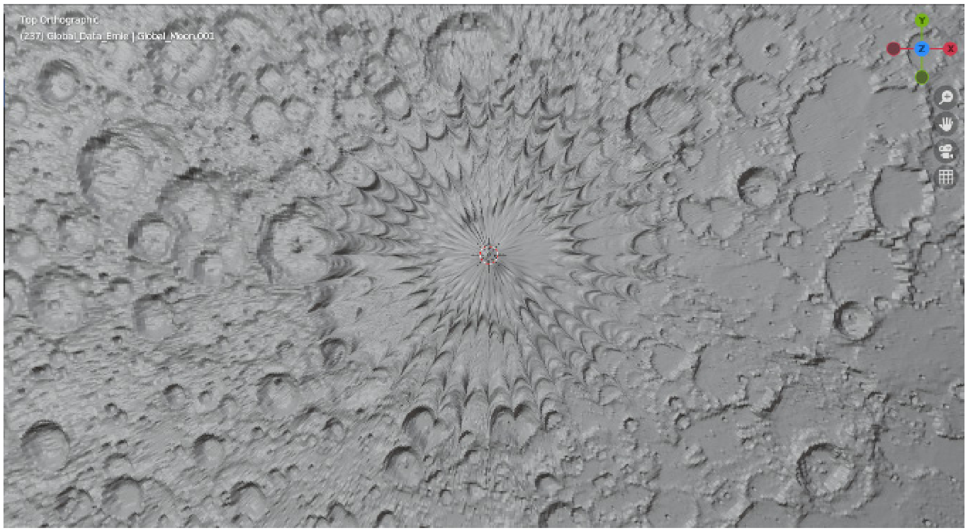


Figure 1.—Lunar terrain pinching due to compressed polar data and mapping capabilities.

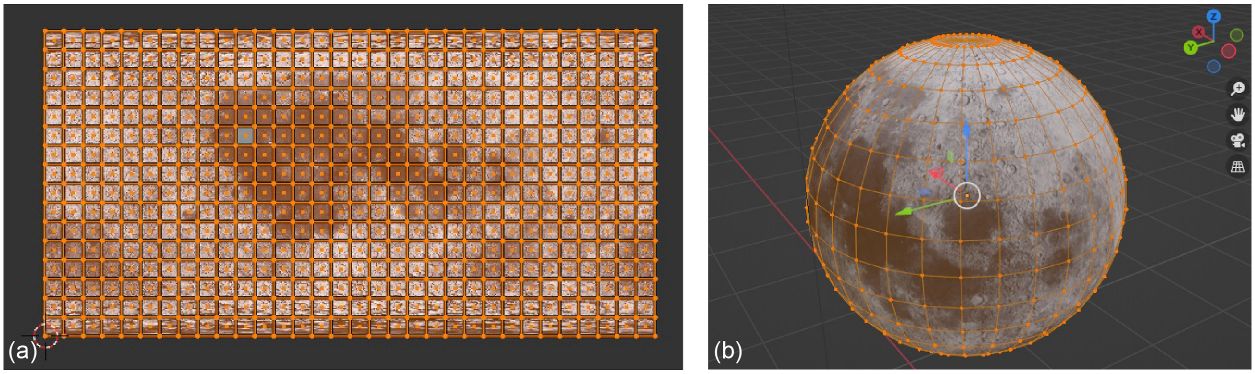


Figure 2.—(a) Tiles along top and bottom of the UV map were reconfigured into rectangular shapes. (b) These rectangular shapes allowed these areas to correspond to mapping of sphere.

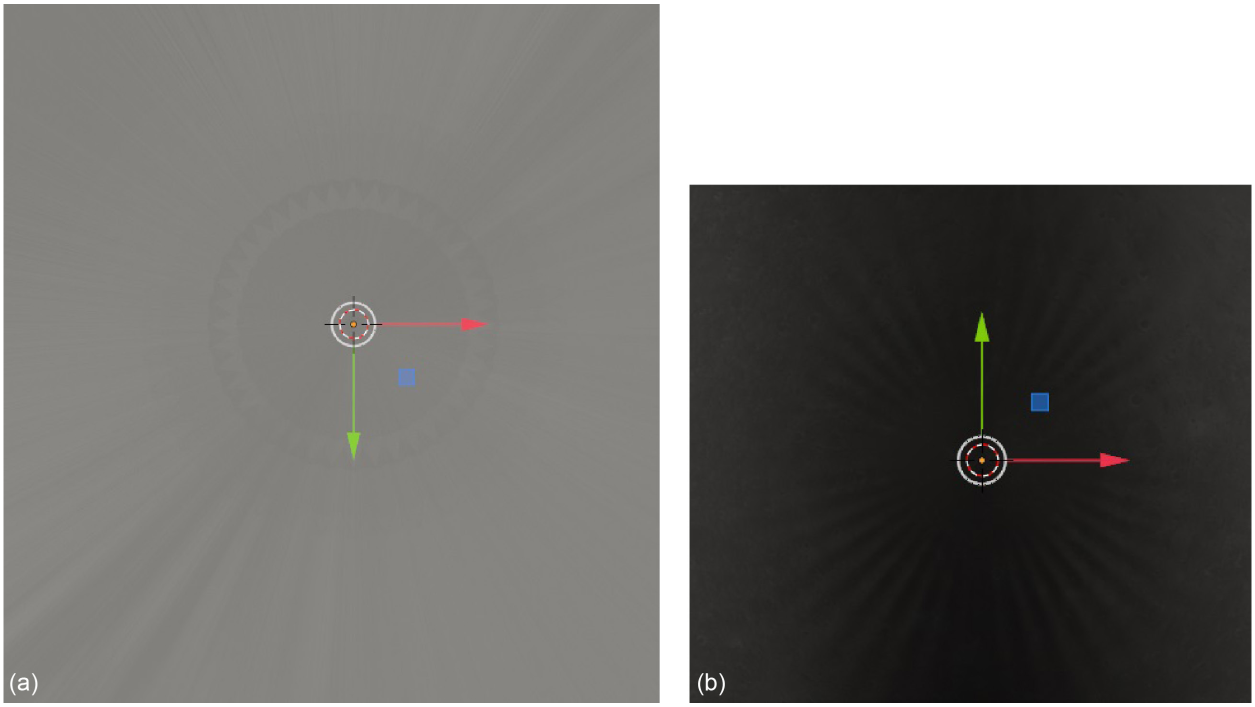


Figure 3.—Occurrences of microwarping. (a) At north pole. (b) At south pole.

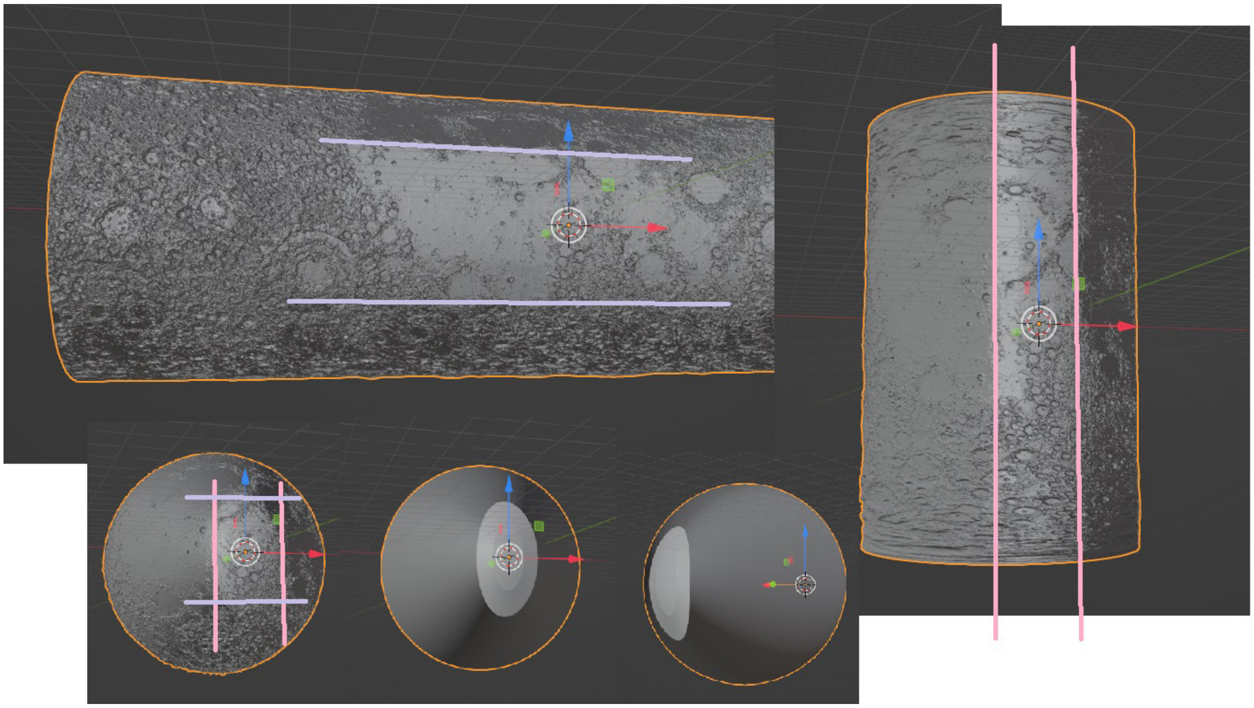


Figure 4.—Two bends of this intermediate attempt to create a sphere from a plane resulted in unwanted artifacts.

Eventually, the development of a process for implementing polar stereographic projection maps into the Moon model rectified polar imperfections and allowed for the implementation of the initial global Moon modeling method, which used a UV sphere for base geometry. After this point, development was redirected towards achieving a new mesh makeup, as shown in Figure 5.

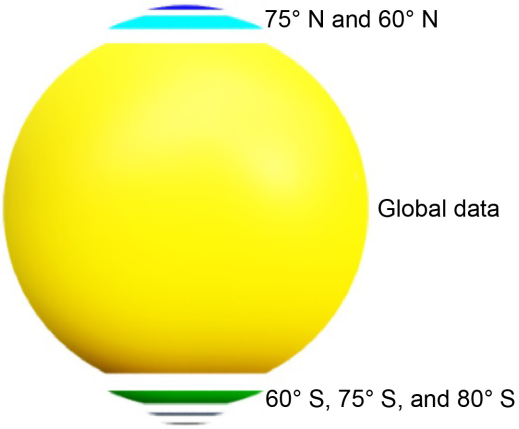


Figure 5.—Basic anatomy of final model.

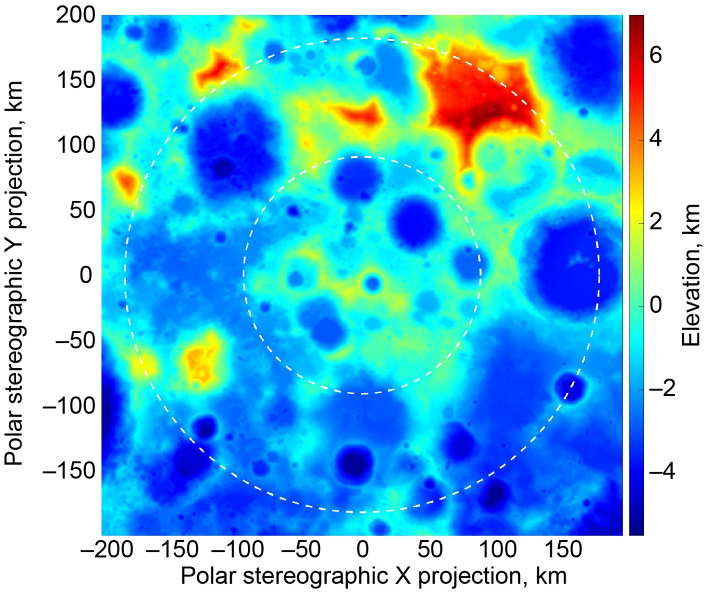


Figure 6.—Example of a polar stereographic map for Artemis region of interest at south pole.

Polar Stereographic Projection Methods

Incorporating DEM polar stereographic maps, such as a grayscale version of the map shown in Figure 6, became vital to the process of visualizing the highest-quality lunar terrain data possible and creating a perfect 3D model of the Moon. One major challenge during this portion of the process was understanding how the polar stereographic maps were created to be able to use them properly within Blender. One form of projection method or another was necessary to adhere square planes representing different polar stereographic projection maps at various latitudes onto a sphere representing the Moon.

Initially, a visually scripted ray-casting formula was used to project the polar stereographic information onto the bottom of the Moon for the southern polar stereographic maps. However, discussion and test cases soon revealed this was not the proper way to project the data onto the model. Although the ray-cast function would project the data, as shown in Figure 7, it would also visually warp the plane incorrectly around the sphere. Essentially, this method compressed data toward the middle of a given image's edges.

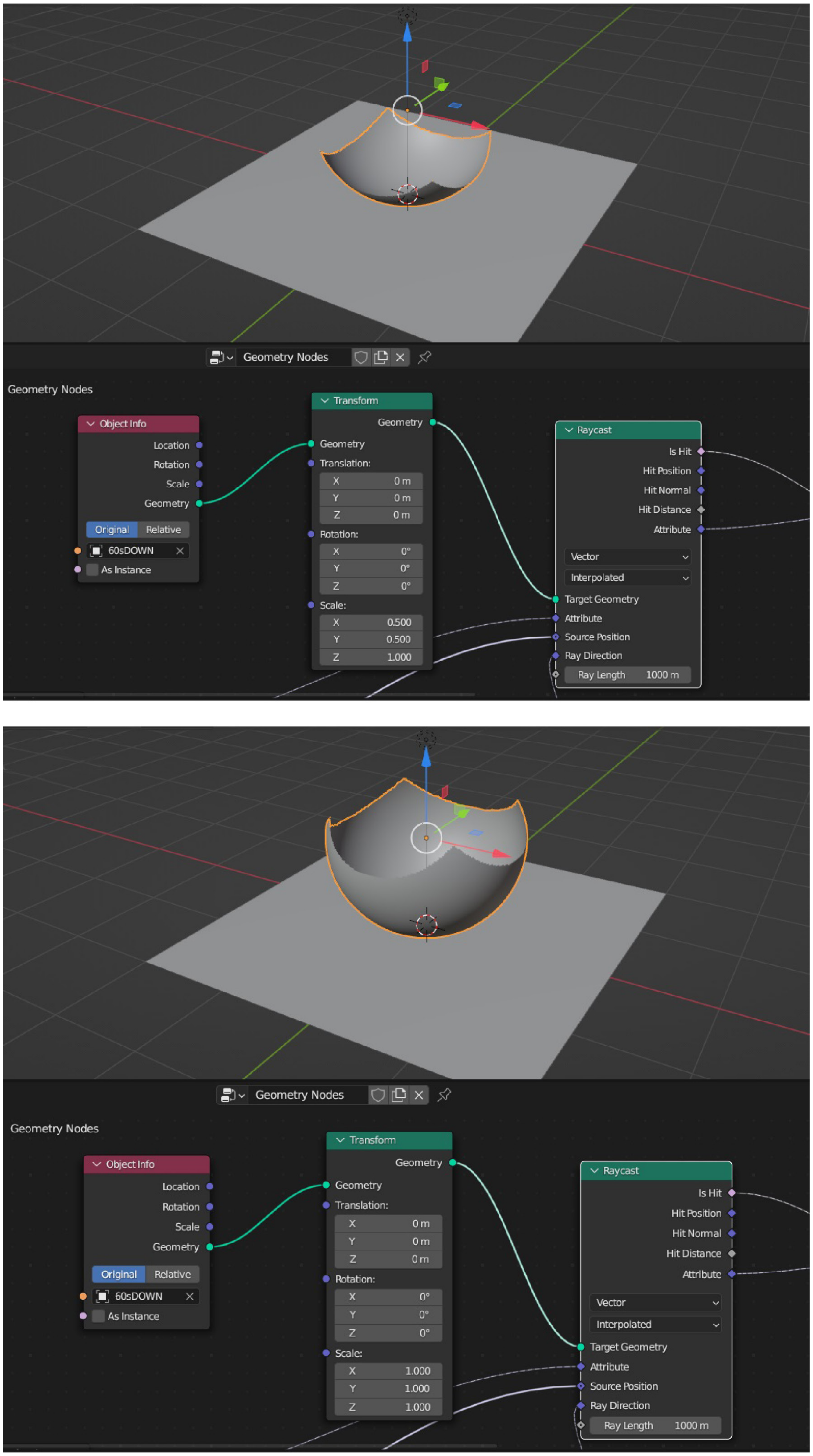


Figure 7.—Scripted ray-casting warps test planes incorrectly, stretching the corners of the image data too far upward.

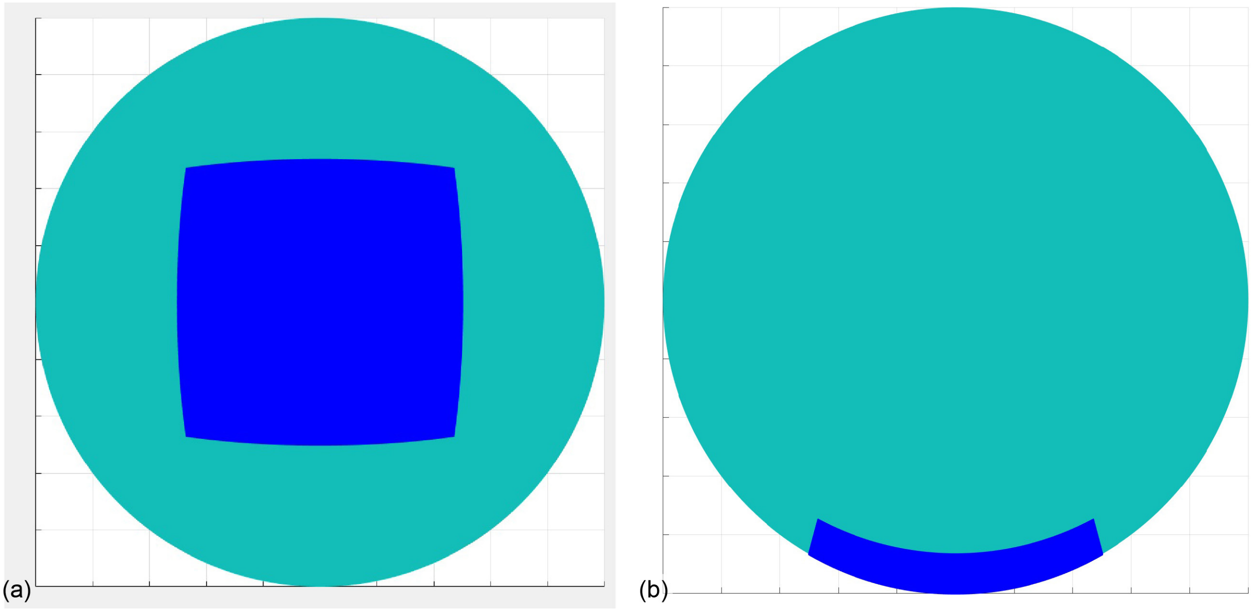


Figure 8.—2D diagrams of mathematically simulated data projection. (a) Bottom orthographic view of proper polar stereographic projection. (b) Side orthographic view of proper polar stereographic projection.

Instead, the development team implemented a simple Project function part of the Shrinkwrap Modifier, which took vertices on the plane and translated them directly upwards along the Z-axis, stopping once the vertices encountered a target sphere. After 2D diagrams of mathematically simulated data projections created using an application developed in MATLAB® programming and numeric computing platform (The MathWorks, Inc.), such as those in Figure 8, were compared with the 3D model and the effect of the projection, this method was selected for a time.

However, when it was the time to combine the refined global and polar stereographic meshes, the lunar terrain texture did not align exactly. There was a concern that the problem originated as an artifact of the global data pinching at the poles. However, referencing the planes to the graphic results of the MATLAB application once again, as well as to a colleague's newly developed lunar representation in three.js, revealed the problem's origin was the shape, size, and bevel of the projection method.

The differences between the plots produced using a MATLAB application and plots developed using Blender, such as those shown in Figure 9, made it clear that our approach to using Blender was incorrect. These results led the authors to theorize that the planar projection results were improperly sized, but further testing was necessary to find a cause and solution.

In addition, it was observed that when considering the entire polar stereographic projection dataset, the shape of the projections within Blender did not correspond with the expected results. This allowed for the visual comparison of the unaltered square shape of the 2D and 3D projection methods. In Figure 10, note how the edges of the projected planes in the Blender 3D software from a side view emulate the perfectly vertical Z-axis, whereas an accurate projection visually presents curved edges that follow the latitude line direction more closely. From the side view, this results in an edge that appears to angle more inward. Both test cases, that is, looking from both the side and the top/bottom, led to the conclusion that the Shrinkwrap Project method resulted in a projection that stretched the data too far at the corners; instead, a correct solution ought to result in smaller dimensions and an outward bevel from a below orthographic view.

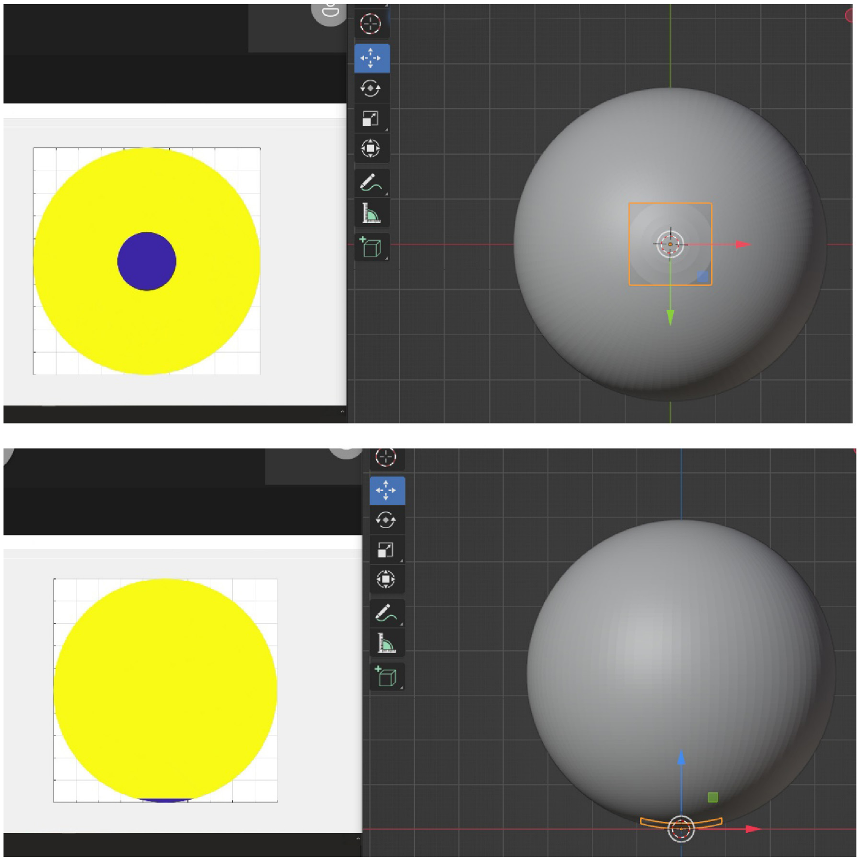


Figure 9.—The sharp contrast between yellow and blue MATLAB-rendered plots and the grayscale plots produced using Blender indicate an incorrect application of Blender for this purpose.

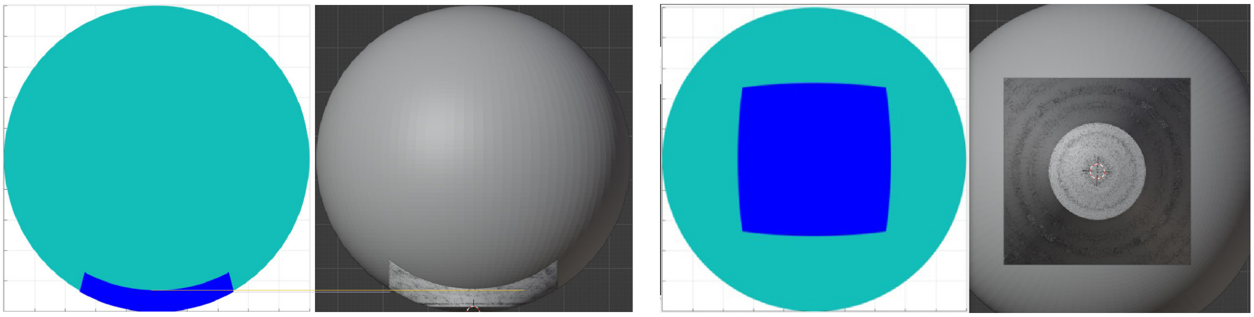


Figure 10.—At 60° South, comparisons are made concerning planar shape predating cutting planes into circular shapes. This illustrates how stereographic projection maps were viewed in Blender with Shrinkwrap Modifier, without constraining data strictly to latitude limit.

The solution was to use the Target Normal projection method in Blender, which projects the plane to the spherical target at the point in which the spherical normal would intersect the plane, coupled with a reformatting of the data, given that Blender wasn't compatible with the nominal polar stereographic projection.

Scaling

Once the general forms of the global data model and the polar stereographic projections were created, the model required additional refinements. Not only were accurate lunar terrain scaling and measurements necessary for a perfect final product, but verifying proper topographic output proved imperative for drawing correct conclusions throughout the process of generating the model and finding the reason for methodology roadblocks.

Early on, Blender's size limitations were encountered when attempting to create large objects to scale, such as the Moon, which is thousands of kilometers wide. Because of this, the Moon's size within Blender was set in meters rather than kilometers in order to work efficiently with various data-dense meshes. As a result, a scale factor of ×0.001 was applied to each measurement when inputting the value. For example, the Moon with a diameter of 3474.8 km would be input as 3474.8 m.

In addition, it was discovered that the Shrinkwrap processes did not in fact accept negative values, despite assertions in Blender's User's Guide that it can handle them. As a result, all DEM datasets were offset by +10 km so there would not be any negative elevation data. This caused the Blender model to use a sphere diameter of 3454.8 m, corresponding to a 10-km reduction from the DEM shift and the ×0.001 scale factor.

Circular Planes

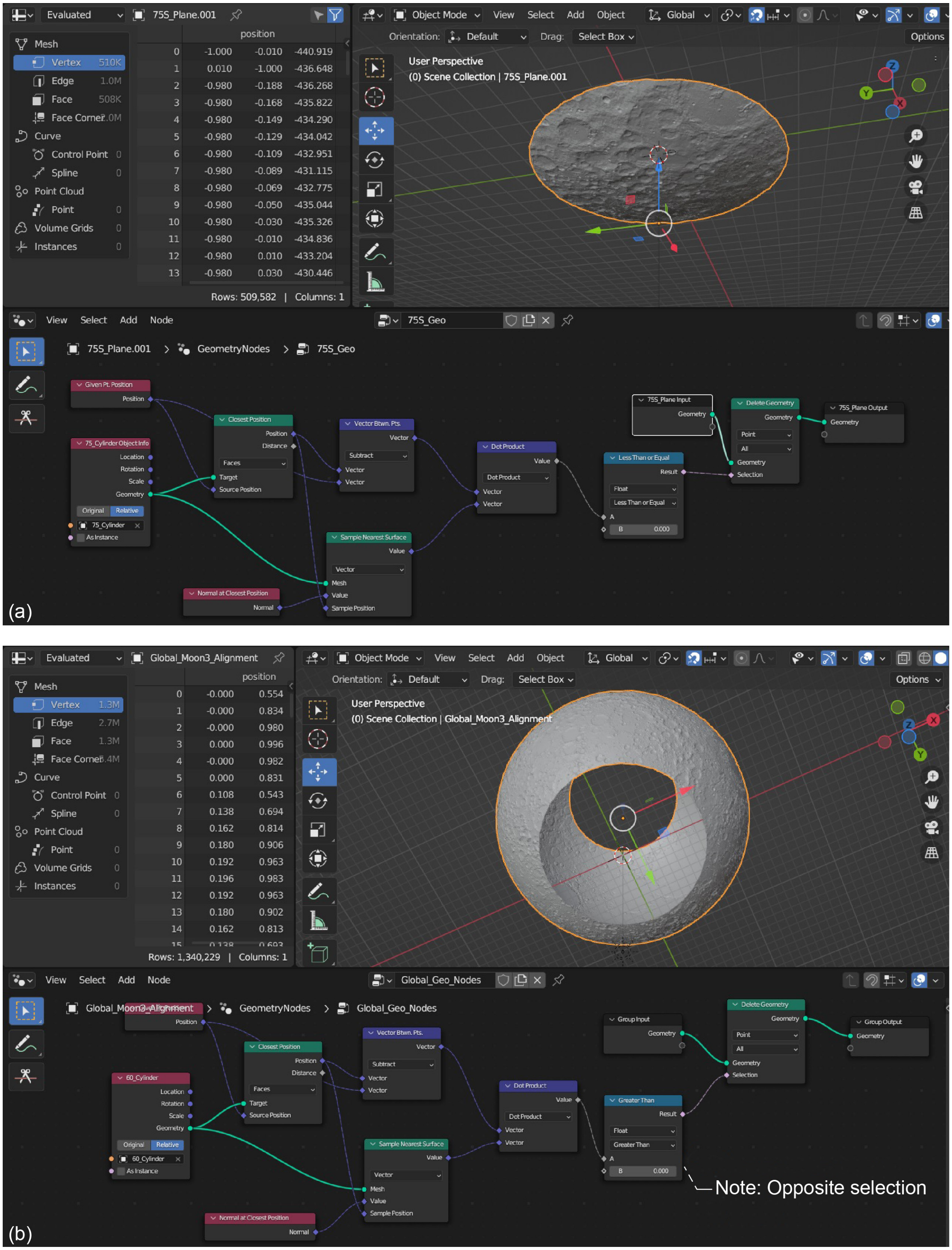
Throughout the research process, the development team experimented with how to modify the square planes, representing various polar stereographic latitudes, into circular planes without deforming the displacement data output. Although the mesh could simply be subdivided to create a circular form upon creating the plane, this would alter the UV map of the plane and most likely conflict with the DEM file input-output methods.

Initially, within the Geometry Nodes visual scripting editor, a Boolean function was used to identify where a new, hidden cube located above the correct line of latitude contacts the plane and deletes the mesh above those points. In this instance, the challenge was working with default Boolean options that recognize mesh overlap.

With this logic in mind, a circular plane was created and projected onto the sphere using the same dimensions and projection as the square planes containing the DEM data. A similar Boolean modifier deleted points that did not overlap with this new, circular plane. With this attempt, complications emerged relating to positioning the circular plane exactly where it needed to be for the function to pick up on this intersection, potentially due to unaligned vertices and faces on a microscopic level.

As a simpler solution, an image of a white circle with a black background was input into the alpha opacity channel on the mesh's principle shader, which controls the material's output. Although this worked to mask out the exterior corners of the square plane, concurrent research and tests regarding export compatibility revealed this method is not yet supported for export.

At this point, the development team revisited the option of using visual scripting and extraneous calculations. Cylinders set to the projected planes' length and width values populated the scene and were made nonvisible. These cylinders ran vertically through the Moon's north and south polar regions, intersecting with the latitude lines of interest. A formula was developed that uses the direction between any given individual point on the Moon's mesh and the closest face on an input target cylinder's mesh. This vector's value is stored along with the normal value of that cylinder's face. The difference between the vector value and the normal is then compared to zero to determine which points are located inside and outside the hidden cylinder. The resulting selection is either kept or deleted from the final output geometry, as illustrated in Figure 11. For each mesh component, the formula was modified slightly to correspond to the correct cylinder(s) and the number of deletions necessary.



75S_Plane.001

	position		
0	-1.000	-0.010	-440.919
1	0.010	-1.000	-436.648
2	-0.980	-0.188	-436.268
3	-0.980	-0.168	-435.822
4	-0.980	-0.149	-434.290
5	-0.980	-0.129	-434.042
6	-0.980	-0.109	-432.951
7	-0.980	-0.089	-431.115
8	-0.980	-0.069	-432.775
9	-0.980	-0.050	-435.044
10	-0.980	-0.030	-435.326
11	-0.980	-0.010	-434.836
12	-0.980	0.010	-433.204
13	-0.980	0.030	-430.446

Rows: 509,582 | Columns: 1

Global_Moon3_Alignment

	position	
0	-0.000	0.554
1	-0.000	0.834
2	-0.000	0.980
3	0.000	0.996
4	-0.000	0.982
5	0.000	0.831
6	0.108	0.543
7	0.138	0.694
8	0.162	0.814
9	0.180	0.906
10	0.192	0.963
11	0.196	0.983
12	0.192	0.963
13	0.180	0.902
14	0.162	0.813

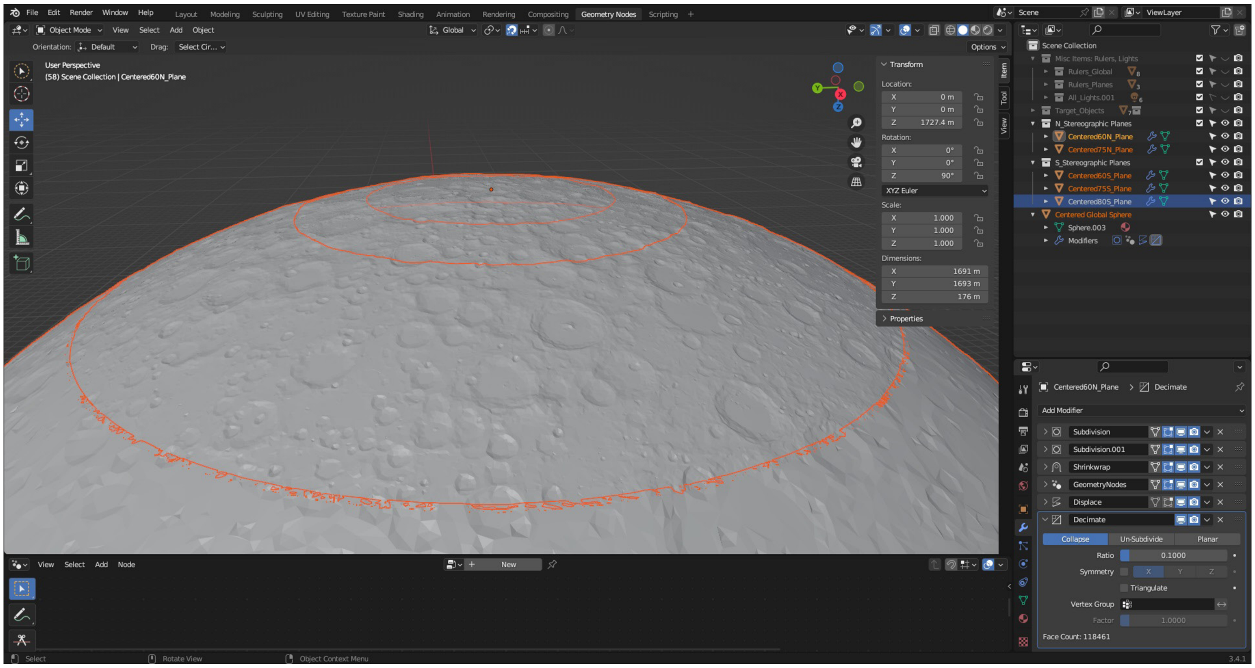
Rows: 1,340,229 | Columns: 1

Note: Opposite selection

Figure 11.—Visual scripting targets and deletes mesh existing outside a hidden cylinder, resulting in a circular plane. By modifying comparison node, opposite selection can also be used. (a) Retained geometry. (b) Removed geometry.

The intersection of meshes required that two hidden cylinders be used to “cut” the meshes on each side of the intersection. Given that the meshes used different resolution terrain data, a common cylinder dimension could not be used because that would create a gap at the intersection of the meshes. As such, cylinders of slightly different sizes were used to overcome the gap created by the terrain resolution differences, so that the combined view of the meshes would not exhibit any gaps. The global to 60° mesh intersection used a 0.5° overlap, while all other mesh intersections used a 0.1° overlap. This intersection, in the lunar south pole view, is illustrated in Figure 12.

Although the cylinder Geometry Node route was applied as a final solution and worked well with the experimentation and procedure for combining the meshes, any of these attempted avenues have the potential to function as desired if explored further, perhaps through combining these ideas of Boolean functions, vector comparisons, and/or projection. In particular, the decision to use cylinders as a basis for incising the meshes consistently was chosen based on known, previously determined measurements of projected polar stereographic maps. If the need arose, and the latitude lines’ height measurements were calculated on an augmented scale of 0 to 3454.8 km, there is no reason why the same concept and logic with a cube resting on or below the desired line of latitude could not produce the same desired outcome.



Orientation: Default
Drag: Select Cir...

Transform

Location:
X 0 m
Y 0 m
Z 1727.4 m

Rotation:
X 0°
Y 0°
Z 90°
XYZ Euler

Scale:
X 1.000
Y 1.000
Z 1.000

Dimensions:
X 1691 m
Y 1693 m
Z 176 m

Centered60N_Plane > Decimate

Ratio 0.1000
Symmetry ☐ X Y Z
☐ Triangulate
Vertex Group
Factor 1.0000
Face Count: 118461

Figure 12.—Lunar south pole meshes shown as being cut by pairs of cylinders and viewed together.

Final Products

After the texture (color) maps were projected in the common process as DEM maps and added to the model, the combined set of six meshes were exported to the .glb file. Figure 13 shows the near-Earth perspective of this export of the model; Figure 14 and Figure 15 show the lunar north and south pole perspectives, respectively. It should be noted that the texture maps are sourced from the same reference of the DEMs used in this work (Ref. 3), as well as the stated interpolation in color map source data above 70° latitude. In Figure 14 and Figure 15, note that the pinching around the poles is no longer present, illustrating one of the major solutions created in this effort.

Furthermore, successful completion and incorporation of the research and modeling carried out in Blender, as seen in Figure 16, has the potential to produce visually powerful and useful imagery for the GCAS by supporting a third-person perspective of the Moon.

Figure 13.—Near-Earth perspective of exported model.

Figure 14.—Lunar north pole perspective of exported model.

Figure 15.—Lunar south pole perspective of exported model.

Figure 16.—Renderings of Moon’s surface. (a) Third-person perspective produced with global data. (b) First-person perspective created using polar stereographic information at 80° South.

Concluding Remarks

The ability to study the Moon’s surface features and evaluate its terrain at various locations will be essential to returning humans to the Moon safely as part of upcoming Artemis missions. The visualization tools in the Glenn Research Center Communication Analysis Suite (GCAS) were developed to help scientists and engineers understand both lunar terrain data and spatial communication information. In the future, the GCAS has the potential to expand to other space exploration missions. Incorporating more planetary bodies into GCAS will support expanding the software’s scope to Mars and beyond.

References

1. Blender: GitHub. https://projects.blender.org/blender/blender.git Accessed May 4, 2023.
2. Blender: 3.5 Manual. https://docs.blender.org/manual/en/latest/ Accessed May 11, 2023.
3. NASA Scientific Visualization Studio: CGI Moon Kit. 2019. https://svs.gsfc.nasa.gov/cgi-bin/details.cgi?aid=4720 Accessed May 9, 2023.