Visualization of Lunar Terrain Isolines Using Dual Marching Squares

Brian J. Stafford\*

University of Illinois at Urbana-Champaign

Abstract

The lunar terrain provides a rich playground in which to explore terrain visualization techniques. The complex geomorphology of the moon’s surface is a result of a number of processes, including volcanic activity and impact craters [1]. This gives rise to a large range of elevations as well as many roughly circular isolines. Marching squares is a popular algorithm for visualizing isolines of such terrain; however, there is empirical evidence that dual marching squares generates better approximations to curved isolines. In an effort to compare these techniques, we have implemented dual marching squares and applied both techniques to the lunar terrain.

**Index Terms**: Terrain—Lunar terrain—Visualization—Marching squares—Dual marching squares

# Introduction

Our work makes use of lunar data generously provided by NASA’s Scientific Visualization Studio [2]. This data was originally captured and processed by the Lunar Reconnaissance Orbiter Camera (LROC) and laser altimeter instrument teams. The color map we use was adapted from the Hapke Normalized WAC Mosaic, which is a composite built from over 100,000 Wide Angle Camera images. This color map includes modifications and combinations of three of the seven wavelength bands of the LROC color data, in order to align with what the human eye perceives. We use the 8192x4096 resolution version here, in PNG format. The terrain height map was obtained from the Lunar Orbiter Laser Altimeter team. We make use of the 16 bpp version in uncompressed TIFF format. The source data covers the globe from 70°N to 70°S, with missing latitudes filled in using monochromatic LROC data and albedo information from the laser altimeter.

This work builds upon the curriculum of the Scientific Visualization (CS 519) course at the University of Illinois at Urbana-Champaign, and in particular the material and lab related to marching squares. We were interested in investigating the visual differences between the marching squares and dual marching squares algorithms. Here, we extend the marching squares lab from CS 519 to include an implementation of dual marching squares. We then compare the two techniques using both a synthetic terrain and a portion of the lunar terrain, using the data obtained from NASA.

Our project is available as a Jupyter Notebook at <https://github.com/brianjstafford/CS519-FinalProject>, where you can find additional details and interactive plots.

# Dual Marching Squares

There is empirical evidence that suggests that dual marching squares both generates better approximations to curved isolines,

\* brianjs4@illinois.edu

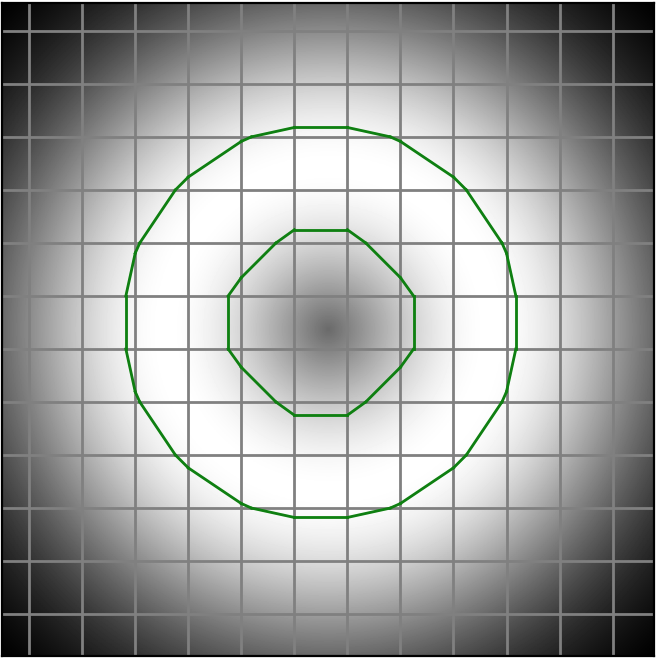
as well as better reproductions of sharp features [3]. These are attractive properties for a visualization technique applied to the moon, with its craggy features and numerous craters. Our initial goal was to visualize isolines on the full lunar terrain (see Fig. 1), but due to performance constraints, we ended up focusing on a subset of the terrain that still allowed us to compare and contrast the different techniques.



1. The full lunar terrain, using height information.

## Implementation of Dual Marching Squares

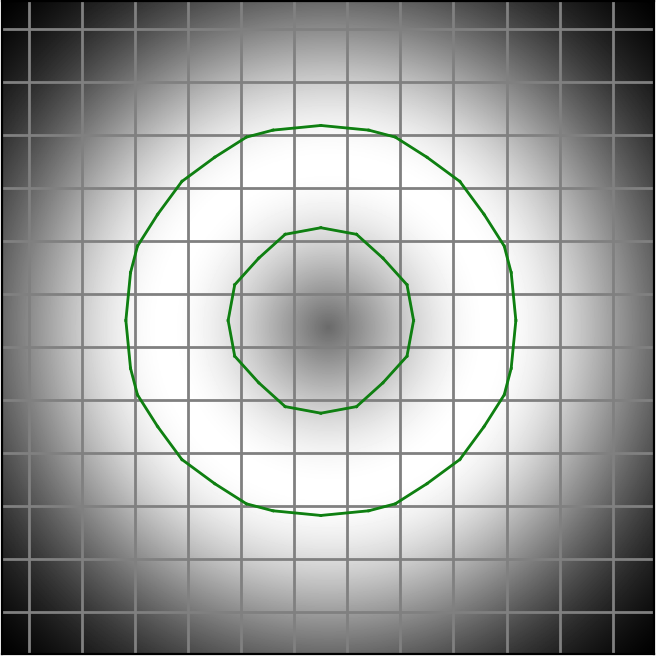
To implement dual marching squares, we built upon the work we already completed as part of the Marching Squares MP in the CS 519 curriculum, following the algorithm provided in [3]. We were able to reuse most of the infrastructural code, including the code to generate the synthetic terrain (consisting of a simple 2-D Gaussian function). We then altered the visualization code to allow the user to toggle between no contours, primal contours (marching squares) and dual contours (dual marching squares). In Fig. 2, we see the results of using marching squares (primal contours). In Fig. 3, we see the results of using dual marching squares (dual contours).



1. Visualizing isolines using marching squares.

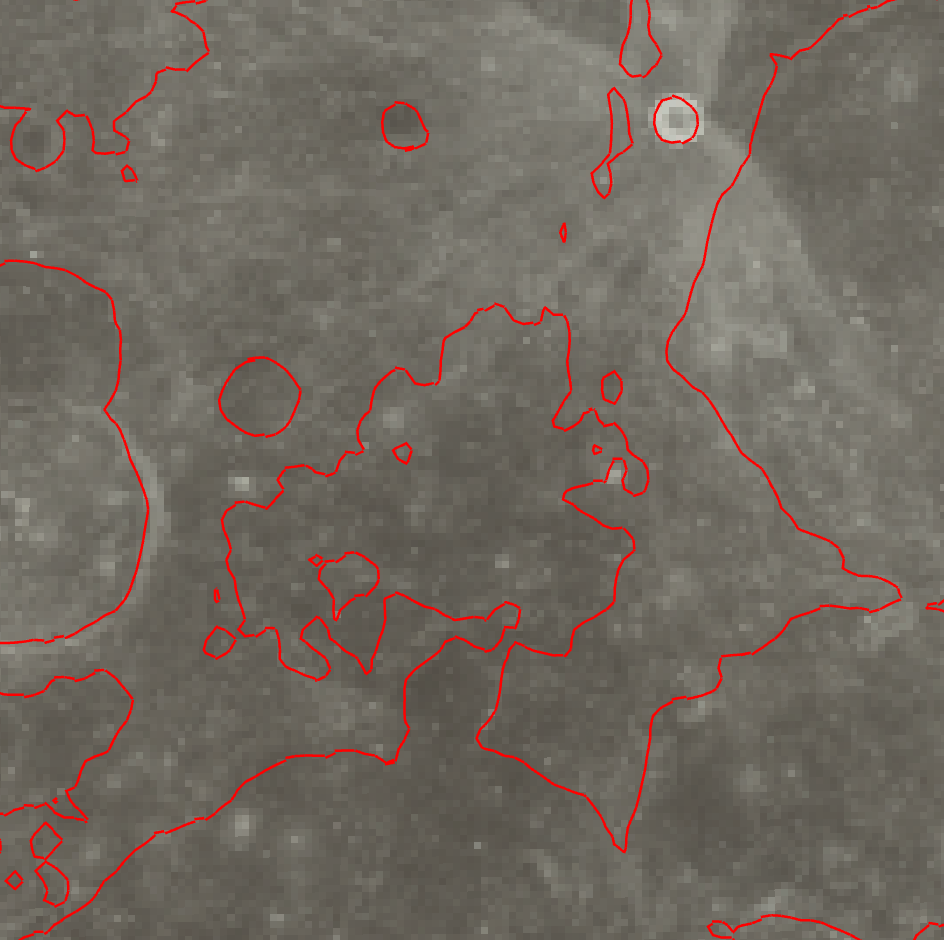
## Applying the Techniques to the Lunar Terrain

Once we completed the implementation of dual marching squares, we turned to applying it to the lunar terrain.



1. Visualizing isolines using dual marching squares.

For performance purposes, we focused on a small subset of the lunar terrain that demonstrated some interesting characteristics. In Fig. 4, we see the isolines (at elevation 0) generated by the built-in contour method in the PyVista visualization library [4].



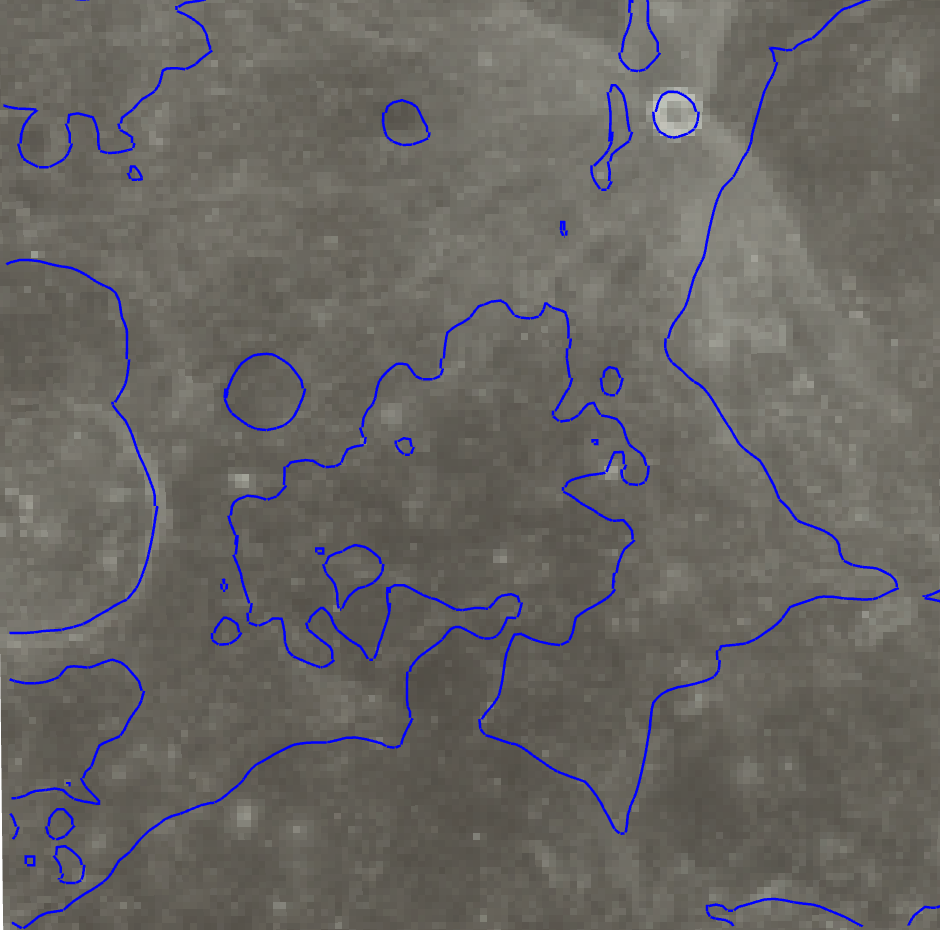
1. Isolines on the lunar terrain generated by marching squares.

In Fig. 5, we see the same subset of the lunar terrain, with isolines generated by our dual marching squares algorithm. In Figures 6 and 7, we show zoomed subsets of these plots, for marching squares and dual marching squares, respectively.

# Discussion

We can clearly see in these figures that dual marching squares provides a better approximation to curved isolines, in both the synthetic terrain from the lab as well as the lunar terrain. When running the Jupyter Notebook, you are able to easily toggle between primal and dual contours on the synthetic terrain, and it’s clear that the dual contours provide better approximations to both circles than those obtained using primal contours. The dual contours do have some sharp points, but those would be significantly reduced with a higher-resolution grid.

The same behavior can be seen in the isolines on the lunar terrain. In Figures 6 and 7, and in the interactive plots in the Jupyter Notebook (where you can zoom in close to see the isolines), it is clear that the isolines generated by dual contours are both smoother than those generated by primal contours and more sensitive to sharp features.



1. Isolines on the lunar terrain generated by dual marching squares.



1. Zooming in on the isolines generated by marching squares.



1. Zooming in on the isolines generated by dual marching squares.

# Conclusion

Dual marching squares is a valuable technique when visualizing terrain such as that of the moon, which includes both sharp and curved features. The implementation we have provided here could be used on additional subsets of the lunar terrain and applied to other types of terrain as well.

In future work, we would like to allow the user to toggle between primal and dual contours when viewing the lunar terrain in the Jupyter Notebook. We would also like to address the performance issues, in order to be able to view an increased subset of the terrain at one time. Finally, we would like to map the lunar color image and the generated contours to a sphere, so the user can interactively view isolines on the moon in a more natural setting.

References

1. <https://en.wikipedia.org/wiki/Geology_of_the_Moon>
2. <https://svs.gsfc.nasa.gov/4720>
3. <https://www.coursera.org/learn/cs-519/home/week/5>
4. <https://docs.pyvista.org/examples/01-filter/contouring.html>