AAS 08-XXX

MACHINE LEARNING BASED CRATER DETECTION FOR TERRAIN RELATIVE NAVIGATION

Chris Gnam*, Timothy Chase Jr†, and Andrew Liounis‡

As Lunar exploration continues to become more commonplace, reliable methods of precise Terrain Relative Navigation (TRN) are needed. While there are many TRN techniques available, one that has received increased interest in the past few years is that of crater based navigation. Crater based navigation has numerous benefits, including being a human recognizable feature (important for crewed missions), as well as the fact that craters are often possible hazards that need to be detected and avoided. The use of crater based navigation has been limited however. This has been due to the difficulty of running such algorithms on board a spacecraft, as well as the difficulty in procuring large amounts of the required training data. This paper presents a new rendering tool for generating large amounts of high quality training data. It then looks at two recently developed machine learning techniques for crater detection and crater identification in real-time on near-future space hardware.

INTRODUCTION

Lunar exploration has become more relevant in recent years, particularly as private industry launches more lunar missions of their own[1] and in collaboration with government partners,[2] and this trend is expected to continue as NASA's Artemis program begins to launch crewed missions. Terrain Relative Navigation (TRN) is a critical part of many navigation systems and thus key to mission success. There are many techniques for TRN, such as template matching which relies on identifying predetermined patches of the target body's surface present in an image. Another approach that has been studied in recent years is that of crater navigation, where rather than arbitrary patches of surface or abstract feature descriptors being identified in an image, a crater can be identified and navigated relative to. Although traditional template matching-based approaches have been favored due to their simplicity, advances in machine learning techniques have made crater detection and identification more feasible for real-time applications.

The deployment of deep learning algorithms onboard spacecraft faces significant challenges due to the unique constraints of the space environment. Recent advancements in computer vision research rely heavily on computationally expensive deep-learning approaches that are incompatible with the limited resources available on spacecraft, primarily due to stringent size, weight, power, and cost (SWaPC) requirements, as well as the need for radiation tolerance.[3] While efforts like NASA GSFC's SpaceCube family of processors, particularly the SpaceCube Low-power Edge Artificial Intelligence Resilient Node (SC-LEARN[4]), aim to bridge this gap, they face limitations such as restricted operational support and reliance on less efficient host processors for certain tasks. Historically, these constraints have limited onboard vision algorithms to simpler template matching techniques for image-space navigation measurements, as recently demonstrated during the landing

*Aerospace Engineer, 595 - Navigation and Mission Design, NASA Goddard Space Flight Center
†Software Engineer, 587 - Science Data Processing, NASA Goddard Space Flight Center
‡Aerospace Engineer, 595 - Navigation and Mission Design, NASA Goddard Space Flight Center

of the Mars Perseverance Rover[5] and the collection of asteroid regolith by OSIRIS-REx.[6] The requirement for radiation-hardened electronics, which typically lag behind consumer-grade hardware in performance, further compounds the issue, and challenges even traditional computer vision algorithms.[7] Additionally, the absence of space-grade GPUs, crucial for advancing terrestrial computer vision models, presents another significant hurdle. Although hybrid computing offers some advantages, it still struggles to meet the computational demands of modern deep learning architectures,[8] resulting in a substantial capability gap between terrestrial and space-based computer vision systems.[3]

This paper first discusses a new rendering tool named Vira, capable of rendering photometrically accurate images, narrowing the gap between training and real-world deployment. It then looks at two machine learning models for performing real-time crater detection and identification, capable of running on near-future space hardware. The ”You Only Crash Once” (YOCO) model is used for crater detection, while the ”Multi-view Attention Regularization” (MARs) model is used for real-time crater identification. Finally, it demonstrates using the new renderer, as well as images from the Lunar Reconnaissance Orbiter (LRO), to train these machine learning models.

VIRA RENDERER

Goddard Space Flight Center has been developing a new rendering tool named *Vira*, after the Inca creator deity, *Viracocha*, the creator of the Universe, the Sun, Moon, and the stars. Publicly available ray tracing software such as the Blender Cycles engine[9] are highly efficient, however, they are typically designed for artists. This means that while they render physically accurate images, they often do not expose all camera or material parameters. Instead, these renderers are intended to be hand-tuned to achieve a specific artistic look. Alternatively, there are tools which prioritize engineering and scientific applications, such as the Goddard Image Analysis and Navigation Tool's ray tracer,[10] which provide the flexibility and control required for certain engineering applications, but often lack modern efficiency developments in ray tracing acceleration data structures.

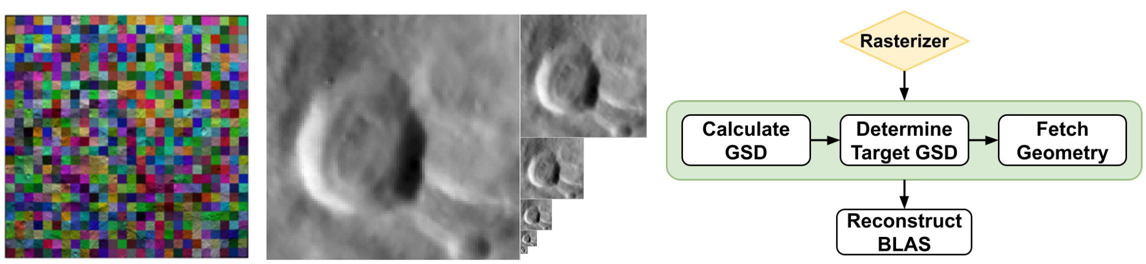
Vira aimed to create a renderer geared towards engineering and scientific applications, while also providing the performance of commercially available rendering tools. Written in C++, it is performant, but provides a high-level Application Programming Interface (API) for configuring scenes with many of the parameters an engineer or scientist may be interested in. These parameters include optical distortion models, aperture size and shape, sensor quantum efficiency, well depth, noise characteristics, and color filter array. Vira uses these to compute the electron counts collected in each photosite of the sensor, which can then be used to generate an image. Users can also cache various steps of the rendering process, which can then be be analyzed or manipulated by hand if desired (such as looking at indirect and direct illumination separately).

Additionally, while many commercial renderers represent light as either a single intensity value, or an RGB value, Vira represents light spectrally. This allows Vira to accurately simulate spectral light sources (such as block body emitters like the Sun), spectral reflectivity of different materials, as well as the quantum efficiency curves of sensors. This is done efficiently through binning of these quantities at compile time, allowing for an approximation of any spectral curve with a user-defined number of bins without needing to perform costly checks or conversions at run-time. This allows Vira to render extremely high quality images very quickly, often rendering faster than similar CPU based renderers such as Blender Cycles. These decreased render times and physically accurate camera and material models make it feasible to generate large amounts of data for training machine learning algorithms.

Level-of-Detail System

There is a tremendous amount of lunar data, with global data sets providing spatial resolution of 60 m across the entire lunar surface, and localized data-sets (such as around the poles) with spatial resolutions as high as 5 m. For many applications (such as simulating images from lunar orbit), it is reasonable to use relatively low resolution datasets, but if you wish to simulate images close to the surface, the highest resolution data sets are required. However, it is infeasible to load all of the highest resolution datasets due to memory limitations, and so they must be carefully chosen based on what is required at any given moment in time. To do so manually can be time consuming for developers whose efforts could be beter focused elsewhere.

To solve this, Vira provides a level-of-detail system to automatically fetch only the data that is required to render an image with pixel-level-accuracy. That is, it will forego loading geometry that does not have an impact on the final rendered image. This is achieved by first pre-processing a Digital Elevation Map into smaller tiles, and then computing a series of subsampled versions of each tile (known as image pyramids). This data is then written to a binary file called a ”Quipu”, which is structured such that any level-of-detail can be quickly loaded. When loaded, these tiles are converted into mesh geometries composed entirely of triangles. Upon startup of the program, the lowest resolution of each tile is loaded, and the current camera pose is used to determine what level-of-detail is required for each tile. Vira ensures that a level-of-detail is chosen such that each triangle is smaller than N pixels (where N is user configurable). Additionally, Vira will load geometry that may not be directly visible to the camera, but which could cast shadows onto the visible portions of the scene.



(a) (b) (c)

Figure 1 (a) Digital Elevation Map split into smaller tiles, (b) An image pyramid generated from an individual tile, (c) the pipeline used for loading the required geometry.

CRATER DETECTION

You Only Crash Once (YOCO)

In the context of space exploration and autonomous spacecraft operations, the accurate detection of surface features, particularly craters, is of paramount importance. Craters serve as crucial landmarks for navigation, landing site selection, and scientific study of planetary bodies. There have been significant developments in real-time object detection algorithms, such as ”You Only Look Once” (YOLO).[11] These perform well when there is an abundance of labeled training data from the domain in which the algorithm will be deployed, resulting in a fully supervised learning approach. However, the challenge of crater detection is compounded by the scarcity of labeled data from diverse planetary environments and the computational limitations of spacecraft hardware. Generating simulated images can yield large amounts of training data, however, even with highly accurate renderers such as Vira there will always be a domain gap between simulation and real-world deployment. These differences are essentially a shift in the distributions that describe the simulated and the real-world environments.

Addressing these challenges, the ”You Only Crash Once” (YOCO[12]) method presents a significant advancement in unsupervised domain adaptive detection of space terrain, including craters. This approach is particularly valuable for in-situ detection scenarios where real-time processing is critical. YOCOv2[13] builds upon its predecessor by enhancing the Visual Similarity-based Alignment (VSA) scheme within lightweight one-stage object detection architectures. While training, YOCOv2 is provided with labeled simulated craters, as well as unlabeled real-world images of craters, allowing it to learn the shift in distributions between the simulated and real-world environment.

YOCOv2's training functions by incorporating novel elements such as perceptually consistent regularization and robust feature selection, which enable the model to learn domain-invariant properties across various terrain types and crater morphologies. YOCOv2's performance has been rigorously evaluated using both simulated and real-world data, demonstrating substantial improvements in Unsupervised Domain Adaptation (UDA) for surface terrain and, in particular, crater detection.

CRATER IDENTIFICATION

Once a crater has been detected within an image, it must then be identified as a specific landmark in order to be used effectively in TRN. There are many ways in which this procedure could be done.[14] These approaches vary wildly from direct feature descriptor matching to identifying patterns of craters analogous to how star trackers work, all the way to anonymous feature tracking in which detection can be matched against a catalog using only the filter states.[15]

MARs

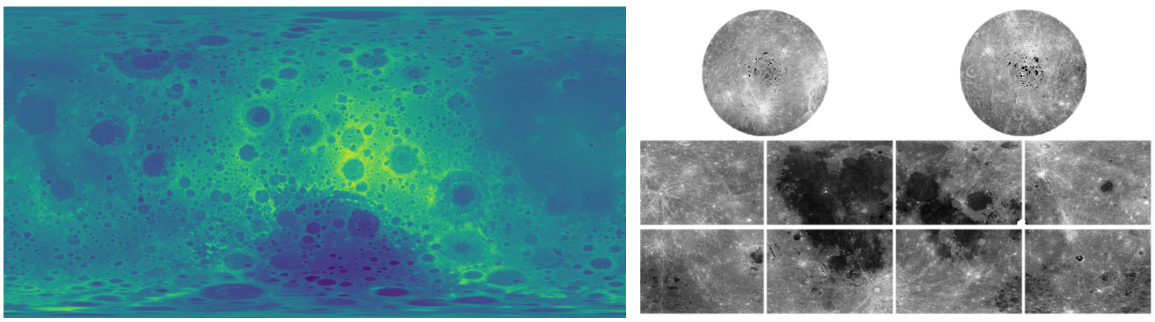
The Multi-view Attention Regularizations (MARs[16]) network addresses the critical need for robust crater description and matching in spacecraft TRN. Current TRN approaches rely on pre-gathered patch-based features for template matching, which are expensive to obtain and limit perceptual capabilities.[6,17] While recent advancements such as YOCO have focused on in-situ detection methods to enhance navigation autonomy, there remains a significant need for a robust description of detected features, especially craters. MARs provides a robust and lightweight description mechanism for space terrain landmarks, focusing on challenges in inter-class similarity and multi-view observational geometry. MARs constrains the channel and spatial attention across multiple feature views, effectively regularizing both the “what” and “where” of attention focus to ensure robust matching under varying illumination and viewing perspective differences. This novel approach aims to improve the robustness of crater description and matching, which is crucial for accurate and reliable TRN.

TRAINING

In order for Vira to be used to render training data, it needs both high-quality geometry and albedo data of the moon. In addition, to generate labels for the rendered images, the locations of craters in each image must be known.

To generate the labels for the training data, we used the Global Database of Lunar Impact Craters created by Stuart J. Robbins.[18] It contains nearly 1.3 million lunar impact craters, providing for each crater both positional information, as well as an ellipse fit to the rim. For each crater in the database, the fit ellipse was projected into the image space of the camera. If the projected ellipse was larger than 5 pixels in diameter, a bounding box was computed and used as a crater label.

For the source geometry and albedo, we used data products generated from the Lunar Reconnaissance Orbiter (LRO). Specifically, the 128 pixels per degree (ppd) Digital Elevation Map (LDEM128) generated using data from the Lunar Orbiter Laser Altimeter (LOLA) was used to generate the 3d surface of the moon.[19] The 643 nm wavelength (304 ppd) Wide-Angle-Camera (WAC) empirically normalized mosaic[20] was then resampled onto LDEM128 to provide albedo values for every point. These two data-sets are shown in Figure 2. This resampling was done with Vira, and then the combined data-set was processed into the Level-of-Detail Quipu files described previously.



(a) (b)

Figure 2 (a) LOLA 128 pixels-per-degree Digital Elevation Map, (b) WAC Empirically Normalized Global Mosaic

Training YOCO

We trained the YOLOv8[21] architecture in supervised (YOLO) and domain-adaptive (YOCO) settings on 5000 Vira rendered images with a resolution of 512x512, where the camera was randomly placed above the lunar surface with a random altitude between 100km and 1000km. For both models, only the Nano (N), Small (S), and Medium (M) sized networks were trained, as these can achieve inference times under one second on a near-future spacecraft processor (Arm Cortex-A9 processor).[13] Examples of the vira rendered and labeled training images can be found in Figure 3.

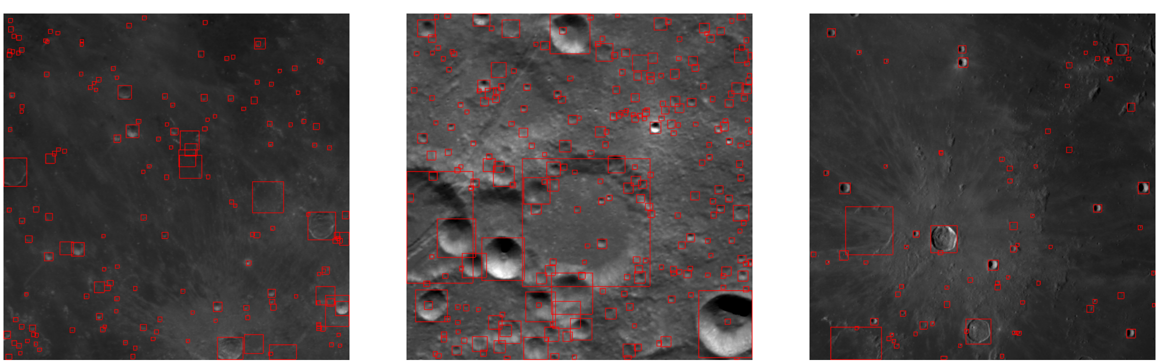


Figure 3 Examples of Vira rendered and labeled training images

Training MARs

MARs was trained on the Luna-1 dataset,[16] which is a collection of crater instances derived from the Lunar Reconnaissance Orbiter (LRO). It was then evaluated against images generated by Vira to demonstrate the ability of the network to identify craters between real-world and rendered images. Although in practice this process would be done in reverse, where MARs would be trained on rendered images and then evaluated on real-world images, the performance should remain the same.

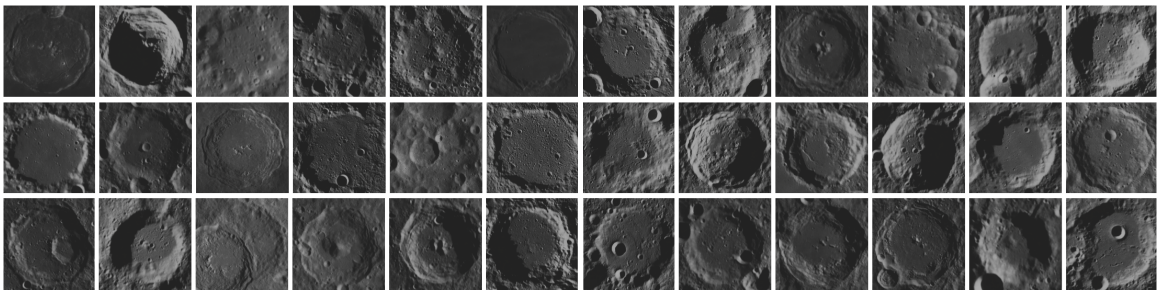


Figure 4 Examples of LRO images from the Luna-1 dataset

Full training and testing pipeline

Figure 5 shows an overview of the entire training pipeline and how the resulting trained networks might be deployed operationally.

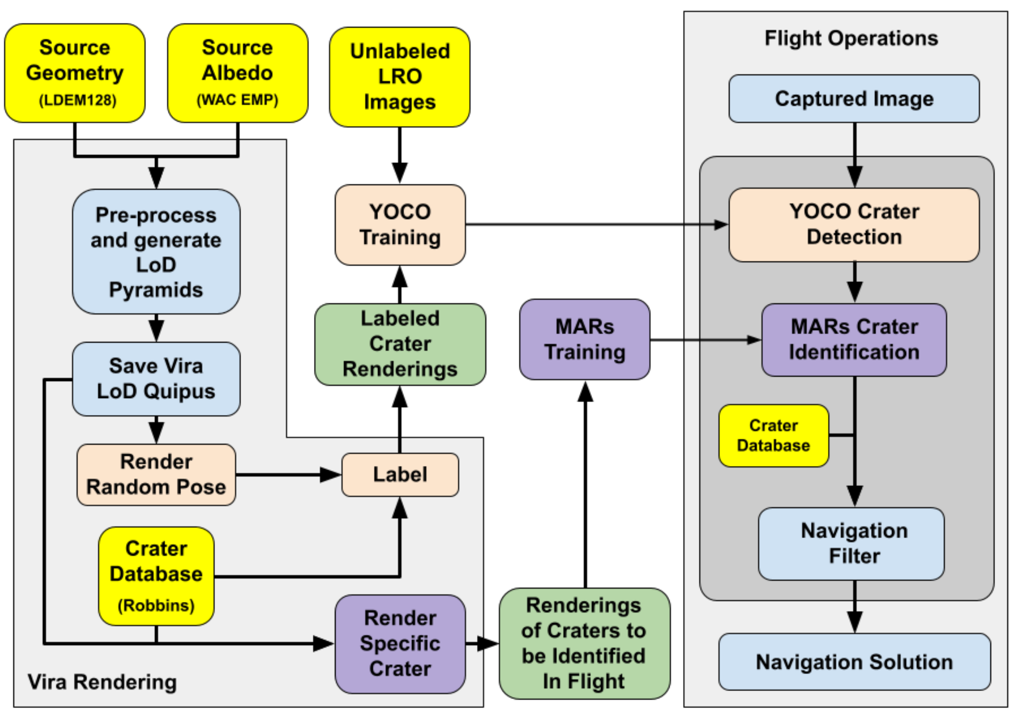


Figure 5 Block diagram of the training pipeline and operational usage

CRATER DETECTION VIA YOCOV2

Figure 6 and Figure 7 show results comparing both YOLO and YOCOv2, for the Nano (8N), Small (8S), and Medium (8M) sized networks. While both performed well, YOCO had fewer false detections, particularly in the Nano-sized network.

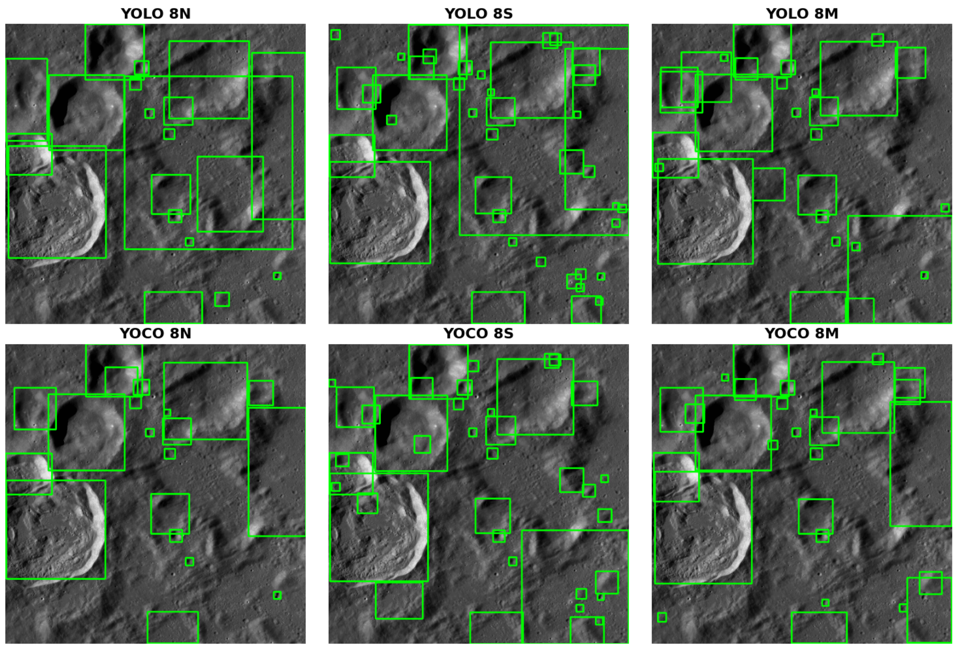


Figure 6 Comparison of YOLO and YOCOv2

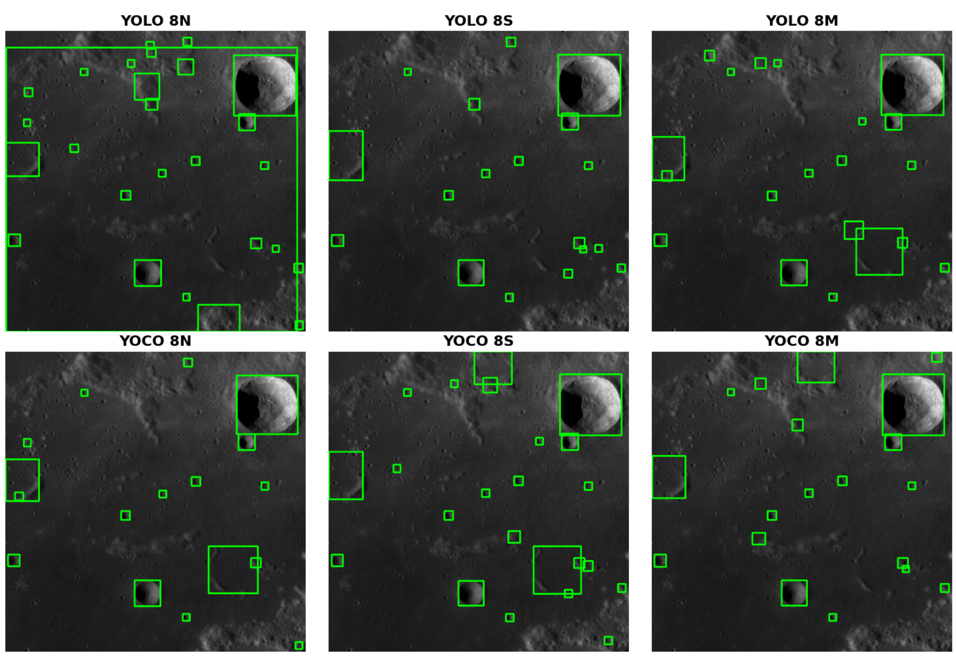


Figure 7 Comparison of YOLO and YOCOv2

CRATER IDENTIFICATION VIA MARS

Figure 8 shows craters from the Luna-1 dataset identified in Vira-rendered images using MARs. The first column is Vira-rendered images cropped to a YOCO-detected crater, while the remaining images are the highest-scoring recalls from the Luna-1 dataset via MARs description, sorted sequentially left to right where a green border represents the true match. The identification was able to recall the correct match and is reasonably robust to changes in perspective, scale, and illumination.

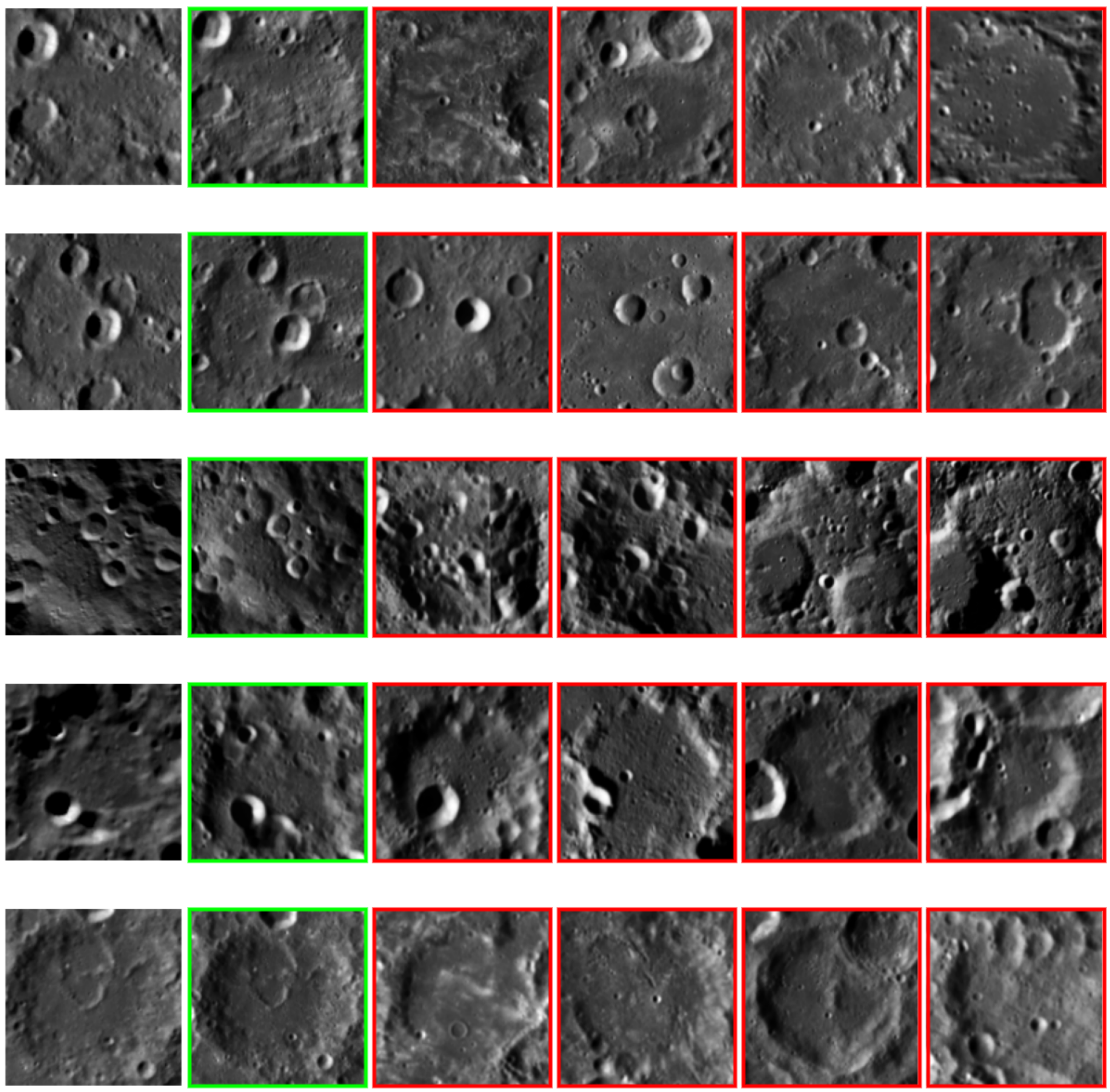


Figure 8 First column: Image craters rendered by Vira. Second Column: Crater from Luna-1 dataset identified by MARs. Remaining Columns: Additional candidates identified by MARs

CONCLUSION

This work demonstrated the usage of a new tool named Vira to efficiently generate high fidelity image renders of the lunar surface. Vira was used to generate and label images to train both YOCOv2 for crater detection, as well as MARs for crater identification against LRO images. The results are promising, as this setup can be used not just to support future lunar missions, but could also be applied to missions around other celestial bodies as well.

Future Work

While both YOCOv2 and MARs demonstrated great performance, if a false detection or crater ID is returned, this needs to be identified before it is ingested by a navigation filter as that could quickly lead to filter divergence. This could be done by using apriori knowledge of the camera pose and crater positions to predict statistical regions in the image where specific features are expected to be.[7] Additionally we would like to explore this work for other geologic features beyond just craters. This is critical for small body missions, as many small bodies do not have many identifiable craters but have other unique geologic features such as boulders.

There are also ongoing developments within Vira to add procedural terrain capabilities. The location of such inserted features would therefore be known exactly, and could be used to generate additional training data for environments where data is currently lacking. Figures 9 and 10 show these experimental capabilities currently under development.

Finally, while Vira has been qualitatively tested, a rigorous validation of Vira's radiometric accuracy is still needed. Such a validation is currently in progress, using images collected by the Artemis 1 test flight around the Moon.[22]

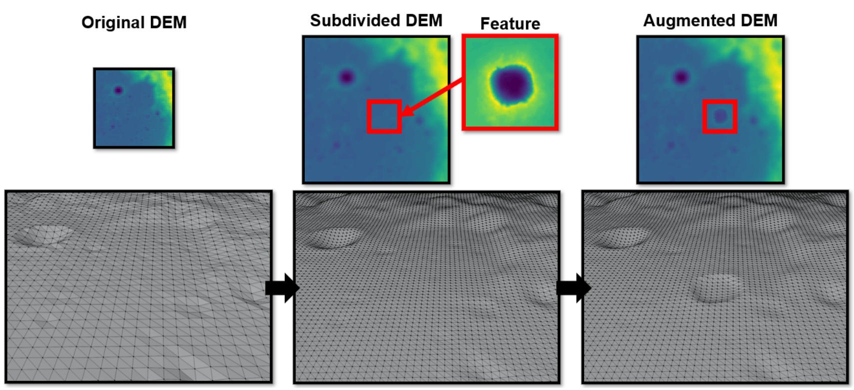


Figure 9 Experimental Vira Procedural Terrain Compositing

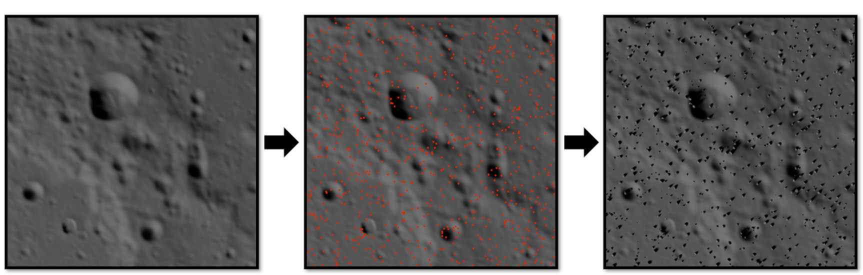


Figure 10 Experimental Vira Rock Scattering

REFERENCES

[1] Astrobotics, *Peregrine Lunar Lander: Payload User's Guide*. Astrobotics.

[2] "FireFly Aerospace Blue Ghost Mission," 2025. `https://fireflyspace.com/missions/blue-ghost-mission-1/`.

[3] T. Chase, J. Goodwill, K. Dantu, and C. Wilson, "Profiling Vision-based Deep Learning Architectures on NASA SpaceCube Platforms," *2024 IEEE Aerospace Conference*, IEEE, 2024, pp. 1–16.

[4] J. Goodwill, G. Crum, J. MacKinnon, C. Brewer, M. Monaghan, T. Wise, and C. Wilson, "NASA SpaceCube Edge TPU SmallSat Card for Autonomous Operations and Onboard Science-Data Analysis," *Proceedings of the Small Satellite Conference*, No. VII-08 in SSC21, AIAA/USU, 2021.

[5] A. E. Johnson, S. B. Aaron, J. Chang, Y. Cheng, J. F. Montgomery, S. Mohan, S. Schroeder, B. E. Tweddle, N. Trawny, and J. X. Zheng, "The Lander Vision System for Mars 2020 Entry Descent and Landing," *AAS Guidance Navigation and Control Conference*, 2017.

[6] C. D. Norman, C. J. Miller, R. D. Olds, C. E. Mario, E. E. Palmer, O. S. Barnouin, M. G. Daly, J. R. Weirich, J. A. Seabrook, C. A. Bennett, B. Rizk, B. J. Bos, and D. S. Lauretta, "Autonomous Navigation Performance Using Natural Feature Tracking during the OSIRIS-REx Touch-and-Go Sample Collection Event," *The Planetary Science Journal*, Vol. 3, No. 5, 2022.

[7] C. Gnam, T. B. Chase, K. Dantu, and J. L. Crassidis, "Efficient Feature Matching and Mapping for Terrain Relative Navigation Using Hypothesis Gating," *AIAA SCITECH 2022 Forum*, 2022.

[8] A. Geist, G. Crum, C. Brewer, D. Afanasev, S. Sabogal, D. Wilson, J. Goodwill, J. Marshall, N. Perryman, N. Franconi, *et al.*, "NASA SpaceCube Next-Generation Artificial-Intelligence Computing for STP-H9-SCENIC on ISS," *Proceedings of the Small Satellite Conference*, No. P1-32 in SSC23, AIAA/USU, 2023.

[9] B. O. Community, *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[10] C. Gnam, A. Liounis, B. Ashman, K. Getzandanner, J. Lyzhoft, J. Small, D. Highsmith, C. Adam, J. Leonard, P. Antreasian, and D. Lauretta, "A NOVEL SURFACE FEATURE NAVIGATION ALGORITHM USING RAY TRACING," *Space Imaging Workshop*, 10 2019.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[12] T. Chase Jr, C. Gnam, J. Crassidis, and K. Dantu, "You Only Crash Once: Improved Object Detection for Real-Time, Sim-to-Real Hazardous Terrain Detection and Classification for Autonomous Planetary Landings," *arXiv preprint arXiv:2303.04891*, 2023.

[13] T. Chase Jr, C. Wilson, and K. Dantu, "You Only Crash Once v2: Perceptually Consistent Strong Features for One-Stage Domain Adaptive Detection of Space Terrain," 2025.

[14] J. A. Christian, H. Derksen, and R. Watkins, "Lunar Crater Identification in Digital Images," *The Journal of the Astronautical Sciences*, Vol. 68, 2021, pp. 1056 – 1144.

[15] J. S. McCabe and K. J. DeMars, "Anonymous Feature Processing for Efficient Onboard Navigation," *AIAA Scitech 2020 Forum*, 2020.

[16] T. Chase and K. Dantu, "MARs: Multi-view Attention Regularizations for Patch-based Feature Recognition of Space Terrain," *European Conference on Computer Vision*, Springer, 2025, pp. 219–239.

[17] C. A. Wright, J. Van Eepoel, A. Liounis, M. Shoemaker, K. DeWeese, and K. Getzandanner, "Relative terrain imaging navigation (RETINA) tool for the asteroid redirect robotic mission (ARRM)," *Guidance and Control Conference*, No. 16-084 in AAS, 2016.

[18] S. J. Robbins, "A New Global Database of Lunar Impact Craters >1–2km: 1. Crater Locations and Sizes, Comparisons With Published Databases, and Global Analysis," *Journal of Geophysical Research: Planets*, Vol. 124, No. 4, 2019, pp. 871–892.

[19] "NASA Goddard Space Flight Center, LDEM_128 LRO-L-LOLA-4-GDR-V1.0," 2017. `https://imbrium.mit.edu/BROWSE/LOLA_GDR/CYLINDRICAL/ELEVATION/`.

[20] A. K. Boyd, M. S. Robinson, and H. Sato, "Lunar Reconnaissance Orbiter Wide Angle Camera Photometry: An Empirical Solution," *43rd Annual Lunar and Planetary Science Conference*, Lunar and Planetary Science Conference, Mar. 2012, p. 2795.

[21] R. Varghese and M. Sambath, "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, IEEE, 2024, pp. 1–6.

[22] M. L. Sarafin and L. S. Logan, "Artemis I: Test Flight Buys Down Risk for Humanity's Return to the Moon," *74th International Astronautical Congress (IAC)*, IAC, 2023.