Building Maps for Terrain Relative Navigation Using Blender: An Open-Source Approach

Kyle. W. Smith[1], Nicholas Anastas[1], Andrew S. Olguin[1], Matthew P. Fritz[1], Ronald Sostaric[2], Samuel Pedrotty[2], Teming Tse[2]

[1]The Charles Stark Draper Laboratory

[2]NASA Johnson Space Center

A persistent challenge for vision-based navigation systems that compare imagery to a reference map is generating high quality maps with similar lighting conditions. Image rendering software can be used to apply variable lighting to reference maps or to generate synthetic imagery for test trajectories. While many image rendering software packages are available, with several developed specifically for spaceflight applications, there are often limitations due to cost, image fidelity, or flexibility. In this paper, we demonstrate the use of an open-source image rendering software, Blender, for use in Terrain Relative Navigation (TRN) applications. A scene in Blender was generated based on elevation data and satellite imagery of the region of West Texas used by Blue Origin for the operation of their New Shepard suborbital rocket. The Blender scene was validated by reproducing imagery collected during a flight of New Shepard in October 2020 and was further used to generate reference maps for use by a TRN algorithm on a subsequent New Shepard flight in August 2021. The work was performed under the NASA Safe and Precise Landing Integrated Capabilities Evolution (SPLICE) project, which is focused on technology advancement for precision landing and hazard avoidance. This work aims to lower the cost of entry and generally promote the adoption and advancement of vision-based navigation technologies.

I. Introduction

There is currently significant interest is using vision-based navigation systems on autonomous lunar landers and missions targeting other destinations such as Mars, asteroids, and beyond. This interest has been bolstered by successful demonstrations of these techniques on the OSIRIS-Rex and Mars 2020 missions.[1,2] While there exist numerous approaches for the specific implementation of terrain relative navigation (TRN), a widely used technique is the correlation of features in imagery collected by the spacecraft to a georeferenced image, or map, stored in an onboard database. There are many difficulties associated with this approach, but this work focuses on the critical problem of generating maps that accurately reflect the scene during the time of the mission. The problem of map generation has significant overlap with that of generating test imagery for potential landing trajectories.

Certain target destinations for future space missions, such as the moon, have reasonably high-resolution (<100 m/px) elevation, imagery, and albedo data publicly available.[3] However, many regions of interest have significant terrain relief features (craters, rile, etc.) that cast shadows, which change the appearance of the scene depending on their orientation relative to the Sun. With a target destination that is locally highly uniform in its material composition, like the Moon, these terrain relief features are often the most convenient to correlate with maps. The problem then becomes generating maps based on elevation, albedo, time, and ephemeris information that accurately capture the shadows that would be observed by the spacecraft during the mission. These maps are typically generated using a graphical rendering software as real imagery of the same region under multiple lighting conditions is not always available.

In addition to map creation, the production of synthetic flight-like imagery by rendering software is essential to performing simulated TRN on multiple trajectories with real image processing algorithms as opposed to relying solely on a sensor model for Monte Carlo analysis. This imagery can also be displayed on a monitor and captured by a camera for hardware-in-the-loop testing.[4]

Rendering accurate shadows prior to flight is also necessary for lunar missions due to the relatively low amount of contrast and features when compared to terrestrial imagery. At low altitudes in a terrestrial setting, the primary source of features is foliage and other spatially small features such as roads which have a relatively small amount of terrain relief and therefore shadows. On the moon, the primary source of features are craters which have a large amount of terrain relief and thus a high likelihood of being shadowed with the sensitivity to lunar sun angle documented in several studies.[5]

Several renderers are currently available for generating appropriately shadowed imagery of terrain under solar illumination, with some even specifically designed for spaceflight applications.[6,7] However, renderers can have limitations due to low image fidelity based on rendering techniques (rasterization as opposed to ray tracing, for instance), availability of tunable renderer settings, or simply due to their high cost per license, flexibility, and ease of use. For this work, we considered Blender, which is an open-source software package equipped with the physics-based path-tracing renderer, *Cycles*.[8] Blender has a convenient Python interface that enabled rendering to be automated and coupled to other open-source libraries for data handling and image processing. Geographic Information System (GIS) data was handled using the open-source library *GDAL*.[9] The GIS data was converted to a three-dimensional model using the open-source library *Open3D* before being imported to Blender.[10] With this tool chain using exclusively open-source software, a landscape in West Texas used for the operation of Blue Origin's New Shepard launch vehicle was virtually recreated. Rendered imagery was tested using the Draper-developed Image Based Absolute Localization (IBAL) TRN algorithm on the imagery in both a simulated environment and during an open-loop spaceflight mission.[11]

II. Blender Scene Description

The workflow from raw data inputs to the final imagery products is summarized in Error: Reference source not found. The Blender scene was composed of four main components: a planar primitive surface decimated into a mesh with vertical displacement at each point to represent elevation and a texture overlay of imagery onto this surface, a camera, and a light source modeling the Sun. The scene covered the region of 31º-32º Nx104º-105º W. Both elevation and imagery data were obtained from the United States Geological Survey National Map download tool.[12,13] Elevation data was of 0.333 arcsecond resolution and was represented by a grid of 1081x1081 vertices. Imagery data was of resolution 1.5 m/px and was represented in the scene by 11132x11132 pixels. For reproducing imagery collected from a test flight, a pinhole camera model was used with a focal length of 16 mm and a detector size of 1440x1080 pixels. Separately, the camera was set to an orthographic projection to capture images for map generation. The *Sun* light source option from Blender was used, which models a collimated light source of non-zero angular size. An angular diameter of 0.5º was used. Solar azimuth and elevation angles were computed using the Python library *Astropy*.[14]

Prior flight tests demonstrated the need to render accurate flight lighting conditions within the satellite image database. Additionally, launch times are not guaranteed and could be delayed hours or days due to holds during pre-flight checkout procedures. Therefore, multiple sets of shadowed imagery for a range of potential flight times were generated. The large area that would be viewed during the flight and the need to generate imagery for variable lighting conditions required an automated pipeline that creates a three-dimensional model of the terrain from the georeferenced satellite imagery and elevation data to be built (Figure 1) . Blender was used to load this model and sun angle information. Blender renders the scene by performing ray tracing to accurately recreate the shadows for the model, which are saved to the image through a process called texture baking. Texture baking is a common process used within the gaming and graphics industry to save the shadows into the 3D model itself to optimize runtime. The texture in our case is the original satellite imagery so this process produces a satellite image that includes shadows for the lighting conditions of interest. The imagery can then be fed into the database generation process.

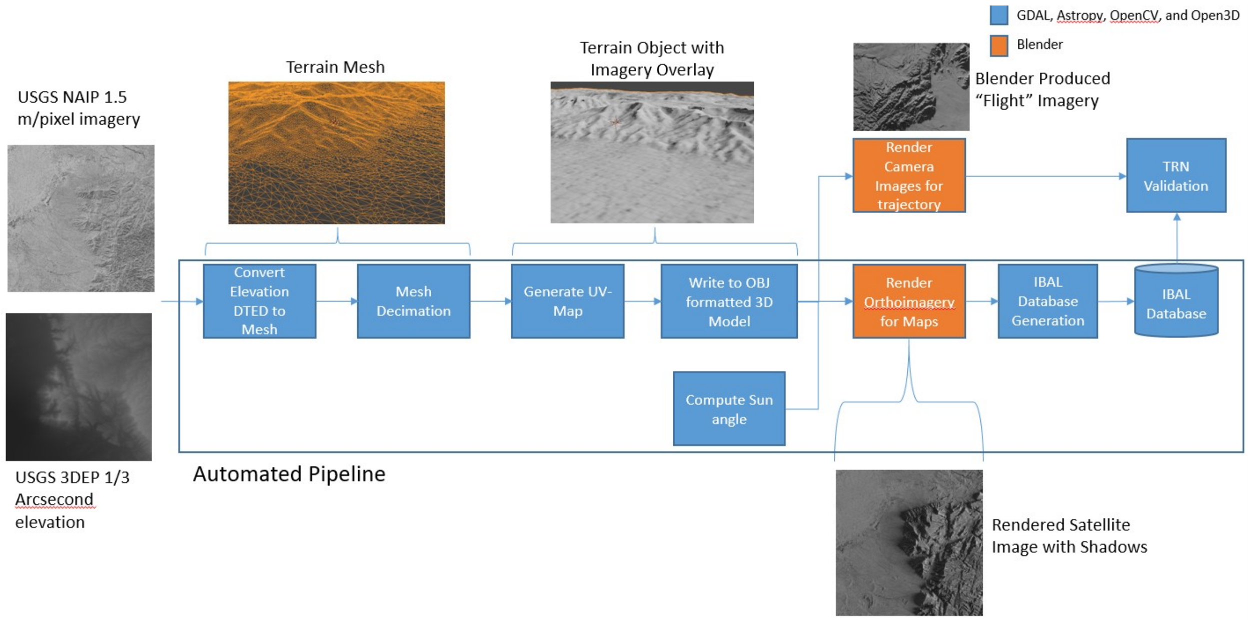

Figure *1. Workflow diagram of map generation and simulated flight imagery with Blender. Data sources from reference* [12].

III. Results

Initial validation of the Blender scene representing the West Texas landscape was performed by reproducing flight imagery collected during a New Shepard test flight on Oct 10, 2020 (Figure 2). The full set of images collected during the flight and other associated sensor data is publicly available.[15] The early flight time of approximately 08:30 Central Standard Time (CST) led to long shadows that produced intricate patterns in the terrain of the mountain ranges to the East and West of the vehicle launch site, both of which were observed during the flight. These lighting conditions were reproduced with high accuracy within the Blender environment. Manual inspection revealed that the rendered imagery captured these shadows with very high fidelity.

Notably, the contrast in the rendered images is greater than what is observed in flight imagery. This is likely due to atmospheric haze that was not captured in the model. Other imaging artifacts include the bokeh present in the bottom and upper-left side of panel A, which was not captured in the model. Further, the renders and flight imagery are only approximately aligned due to uncertainty in the camera position and attitude at each frame. The disagreements noted here were deemed acceptable and the scene was used to generate georeferenced orthoimagery of the area. The output map was then processed into a database of features using methods that are specific to the IBAL code base, but the map could just as easily be parsed for any TRN application at this point in the process.

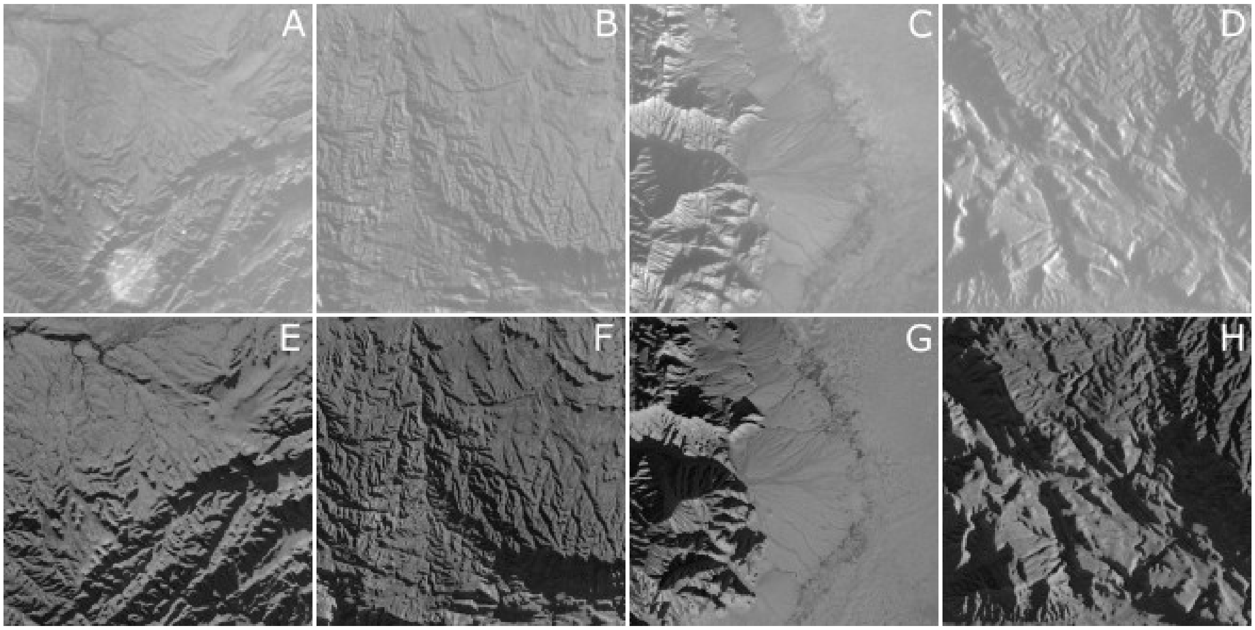

Figure *2. Comparison of imagery collected at various altitudes during the Oct 10, 2020 test flight (A-D) and corresponding renders using an approximated camera pose and camera model (E-H).* Images were cropped to square aspect ratios for display purposes. Images are centered at approximately 31.316ºN, 104.551ºW (A), 31.637ºN, 104.647ºW (B), 31.511ºN, 104.902ºW (C), 31.597ºN, 104.713ºW (D). Raw imagery available.[15]

The test flight data was processed with IBAL using a database generated from only USGS imagery and again using the Blender-generated database. Both databases provided accurate and persistent correlation of non-shadowed features of the terrain (vegetation, soil material differences, etc). However, the USGS imagery database failed to correlate features in the mountainous regions, as the imagery was collected when the Sun was near zenith. Using the Blender generated database, the TRN algorithm identified over 400 unique features comprised of both terrain relief features and albedo features over 400 continuous seconds of the flight with a field of view spanning two different mountain ranges and a valley floor between them (Figure 3).

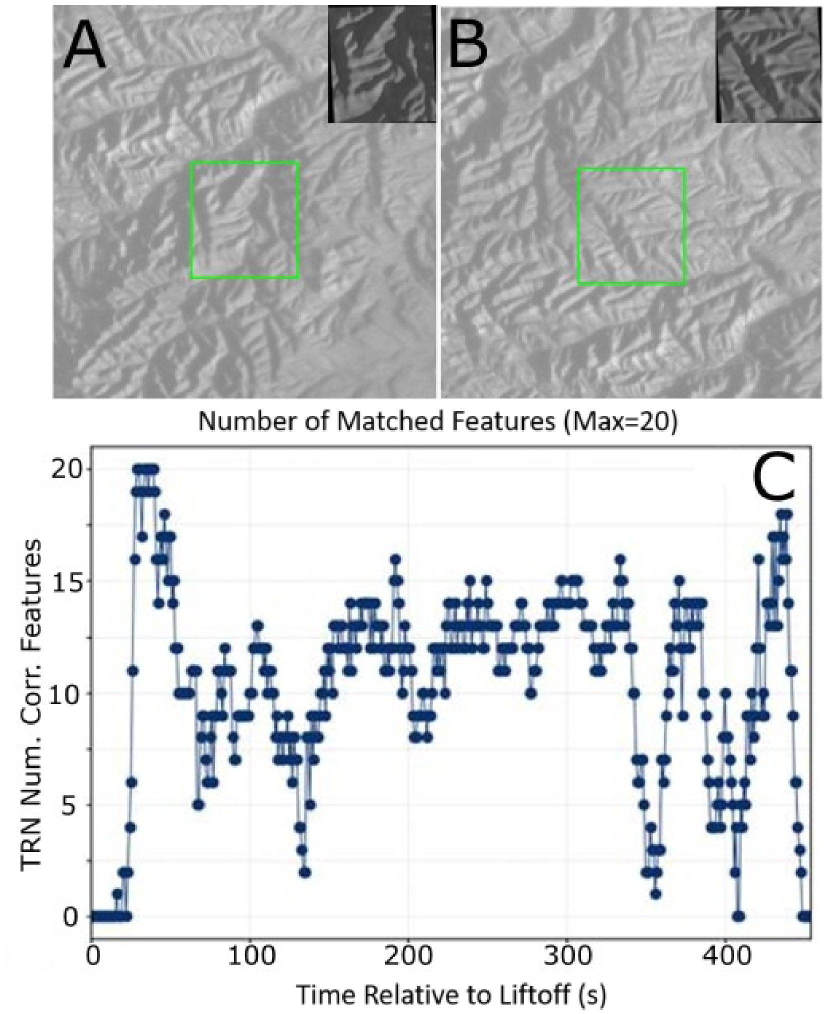

Figure *3. TRN performance using Blender generated maps and imagery from the Oct 2020 New Shepard test flight. Cropped regions of flight imagery with inset features from database (A, B). The inset map feature was "warped" to align with the estimated perspective of the camera. The green square shows the region that TRN identified as corresponding to the map feature. Plot of the number of features matched over the duration of the approximately 450 seconds flight (C).*

Uncertainty in the launch time for a second test flight aboard the New Shepard vehicle led to the generation of multiple maps under different expected lighting conditions. Additional logic was added to the TRN algorithm to only search for features from the database that most closely matched the current time. The databases were binned into two-hour segments, but analysis on the size of the sun angle change required for a new database has not been performed. Both manual inspection and feature selection algorithms, such as corner detection, clearly reveal that there are different optimal features under the illumination conditions for each database (Figure 4).

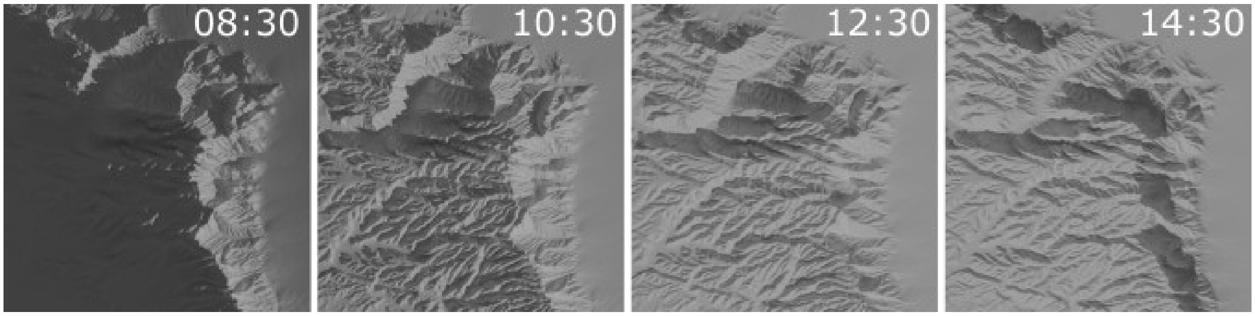

Figure 4. *Segments of the orthographic projection map of West Texas scene rendered under lighting conditions corresponding to 08:30, 10:30, 12:30 and 14:30 local Central Standard Time on Oct 10, 2020. The imagery overlay was removed from these renders to highlight only the differences in terrain relief features. The displayed region is centered around approximately 31.436º N, 104.941º W.*

The second test flight occurred at approximately 09:31 CST/14:31 UTC on Aug 26, 2021 at the same location and with the same approximate trajectory. The TRN algorithm correctly used the 10:30 database for feature matching during the flight. This provides a real-world dataset of imagery collected from a very similar trajectory under distinct lighting conditions. The high altitude, predominantly vertical character of the trajectory and propulsive landing makes the dataset especially useful as an analog to future spaceflight landing missions. The TRN algorithm consistently identified close to the maximum number of features searched for in each frame over the majority of the flight (Figure 5). This was a successful demonstration of maps generated using Blender capturing distinct lighting conditions which were then used by a TRN algorithm.

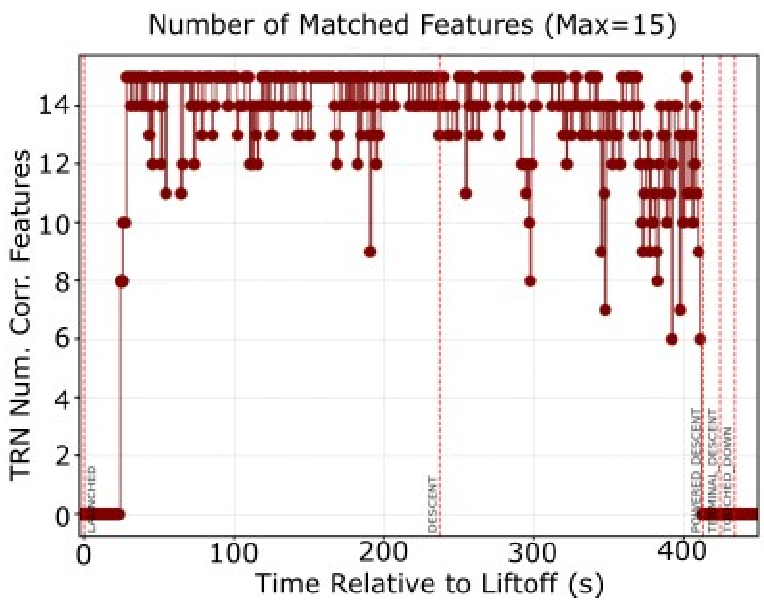

Figure 5: Number of features identified by TRN using imagery from the flight on Aug 26, 2021.

IV. Future Work

Several limitations to this approach have been identified in this work. The more physically realistic ray tracing rendering engine *Cycles* does not offer real-time rendering capability, which therefore limits its use for test imagery generation in a closed-loop context. The performance impact of using real-time, raster engines, such as *Eevee,* on image quality for TRN purposes has not been explored.

Another area to be improved upon is the material properties of the scene. No exploration of the vast tunability of material properties in Blender to obtain a better match with the reflectivity and scattering properties of the landscape was performed in this work. This would likely be critical to lunar scene generation given the optical properties of lunar regolith.

Additionally, the trajectory of New Shepard is primarily vertical and thus only requires a relatively small area of the Earth to be generated as a scene. As such, limitations due to available computer memory were not encountered. Lunar trajectories that traverse large areas would likely need to give this consideration and potentially segment the trajectory into different scenes.

V. Conclusions

Image rendering of environments under different lighting conditions is essential for the use of TRN algorithms on future space missions to shadowed landscapes. We report the use of open-source software packages for physics-based image rendering to reproduce test flight imagery and to generate time-specific shadowed maps for the purposes of developing and testing TRN sensors. Application of this rendering scheme in the West Texas region resulted in successful image-based navigation of a sensor suite aboard a vehicle at altitudes exceeding 100 km. There appears to be no fundamental barriers to this software and methodology being extended to lunar or other environments.

Acknowledgments

The work presented in this paper funded by the National Aeronautics and Space Administration (NASA) through the Johnson Space Center as part of technology advancement and demonstrations for the SPLICE project in conjuncture with the NASA Flight Opportunities Office. Flight testing and verification was made possible by Blue Origin, LLC based in Kent, WA.

References

[1] Lorenz, D. A., Olds, R., May, A., Mario, C., Perry, M. E., Palmer, E. E., and Daly, M. “Lessons Learned from OSIRIS-Rex Autonomous Navigation Using Natural Feature Tracking.” *2017 IEEE Aerospace Conference*, 2017, pp. 1–12.

[2] Cheng, Y., Ansar, A., and Johnson, A. “Making an Onboard Reference Map From MRO/CTX Imagery for Mars 2020 Lander Vision System.” *Earth and Space Science*, Vol. 8, No. 8, 2021, p. e2020EA001560.

[3] Barker, M. K., Mazarico, E., Neumann, G. A., Zuber, M. T., Haruyama, J., and Smith, D. E. “A New Lunar Digital Elevation Model from the Lunar Orbiter Laser Altimeter and SELENE Terrain Camera.” *Icarus*, Vol. 273, 2016, pp. 346–355.

[4] Holt, G. N., D’Souza, C. N., and Saley, D. W. “Orion Optical Navigation Progress toward Exploration Mission 1.” *2018 Space Flight Mechanics Meeting*, 2018, p. 1978.

[5] Manni, C., Mario, C., Ahmed, N., and McMahon, J. “Optimizing Feature Lighting Conditions for Vision-Based Terrain Relative Navigation.” *2nd RPI Space Imaging Workshop, Saratoga Springs, NY*, 2019.

[6] Brochard, R., Lebreton, J., Robin, C., Kanani, K., Jonniaux, G., Masson, A., Despré, N., and Berjaoui, A. “Scientific Image Rendering for Space Scenes with the SurRender Software.” *arXiv preprint arXiv:1810.01423*, 2018.

[7] Martin, I., Dunstan, M., and Gestido, M. S. “Planetary Surface Image Generation for Testing Future Space Missions with Pangu.” *2nd RPI Space Imaging Workshop*, 2019.

[8] Community, B. O. Blender - a 3D Modelling and Rendering Package. http://www.blender.org.

[9] GDAL/OGR contributors. GDAL/OGR Geospatial Data Abstraction Software Library. https://gdal.org.

[10] Zhou, Q.-Y., Park, J., and Koltun, V. “Open3D: A Modern Library for 3D Data Processing.” *arXiv preprint arXiv:1801.09847*, 2018.

[11] Dever, C., Hamilton, L., Truax, R., Wholey, L., and Bergeron, K. “Guided-Airdrop Vision-Based Navigation.” *24th AIAA Aerodynamic Decelerator Systems Technology Conference*, 2017, p. 3723.

[12] U. S. Geological Survey. USGS National Map Downloader.

[13] U. S. Geological Survey. 20201228, USGS 1/3 Arc Second N32w105.

[14] Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., and others. “Astropy: A Community Python Package for Astronomy.” *Astronomy & Astrophysics*, Vol. 558, 2013, p. A33.

[15] National Aeronautics and Space Administration. Deorbit Descent and Landing Flight 1 (DDL-F1). 2021, Retrieved from: Https://Data.Nasa.Gov/Aerospace/Deorbit-Descent-and-Landing-Flight-1-DDL-F1-/Vicw-Ivgd/Data.