

EXTENDING CAPABILITIES OF CRATER NAVIGATION AND TIMING FOR AUTONOMOUS LUNAR ORBITAL OPERATIONS

Rachael E. Gold^{1*}, Sofia G. Catalan¹, Brandon A. Jones² and Renato Zanetti²; ¹Graduate Research Assistant, ²Assistant Professor (The University of Texas at Austin & 2617 Wichita St, Austin, TX 78712). * [rgold@utexas.edu]

Abstract. *Autonomous navigation and timing in the cislunar regime is challenging due to the lack of satellite-based navigation, but imaging of lunar terrain allows for the extraction of positioning information. This paper presents recent advancements in a Crater Navigation and Timing (CNT) algorithm capable of autonomously determining the state of a satellite. Improvements to both the generation of simulated imagery through a crater template-based method and the navigation filter through the addition of measurement bias estimation and nonlinear process noise are displayed. Results of Monte Carlo tests evidence successful state, time and measurement bias estimation.*

Introduction. Operations in the Cislunar regime are of interest to NASA, the DoD, and commercial entities, but the lack of satellite-based navigation makes autonomous navigation difficult. The most readily available positioning information comes from the lunar terrain, or more specifically, craters. The Crater Navigation and Timing (CNT) algorithm seeks to use camera images of craters in order to autonomously determine the translation state in lunar orbit.

Algorithms for crater-based terrain relative navigation in the lunar environment have been developed.¹ Previous methods include fast Fourier transforms (FFTs), shadow modeling and pixel intensity thresholding, but they are limited by geometric and lighting constraints such as crater shape and shadowing. The CNT algorithm uses a machine learning (ML)-based detector, which may relieve the algorithm of most geometric and lighting constraints through the training data. ML-based detectors have been used by others as well^{2,3} with a focus on the lunar landing problem. The CNT algorithm does not attempt to solve the landing problem, rather it enables orbital operations in Low Lunar Orbit (LLO). While the two problems share many aspects, challenges specific to the orbit problem include the necessity to process images at a higher altitude, regular periods of full shadow that do not provide usable imagery, and estimation in the inertial frame rather than the fixed frame. This paper will address improvements made to the CNT algorithm which was presented previously.⁴

The paper will cover changes made to the image generation and processing and changes to the crater identification and measurement model to account for measurement bias and process noise. The most significant changes to the filter are the addition of the estimation of the measurement bias and the choice to represent the measurements in pixel space rather than by bearing angles. The former of the two changes is motivated

by the bias evidenced in the previous paper⁴ when using the Lunar Reconnaissance Orbiter Camera (LROC) mosaics and the expected necessity for measurement bias estimation in operation. The bias in the LROC mosaics also motivated the use of the template-based image generation. The latter of the changes is due to the increased nonlinearity that comes with representing measurements in angles rather than in pixels. The paper will also show the filter performance of the fully integrated algorithm with the changes previously mentioned implemented. The goal for the algorithm performance is to obtain a position error and time error of less than 100 meters per axis and 100 milliseconds, respectively. To evidence progress towards this goal, two Monte Carlo (MC) tests are provided with images in-the-loop and two tests with simulated measurements. Additionally, the mean and standard deviation of the RMS error of nine tests with simulated detections are provided. Some initial results processing images taken by LROC through the detector are presented, while full processing of LROC images through the CNT algorithm is left to future work.

The paper will be organized as follows. First, the implementation of the algorithm will be discussed. This is followed by a description of the template-based image generation process. The test case generation process and a discussion of results are next. Finally, the future work and conclusions are described.

Implementation. The CNT algorithm is split into four parts: crater detections, crater identification, spacecraft state estimation and time estimation using ground-based ephemeris solutions. The position estimation first requires the processing of images through the Mask R-CNN algorithm in order to obtain crater positions in the camera frame. This is followed by the identification of craters using the 2018 Robbins Lunar Crater Catalog.⁵ Finally, the measurements, that correspond to identified craters, are fed into an Extended Kalman Filter (EKF) where the state (position, velocity and measurement bias) is estimated. The EKF includes nonlinear propagation of the process noise transition matrix (PNTM) and estimation of any measurement biases. The time bias is determined through a least squares fit to estimate the time that minimizes the deviations of the navigation solution with a ground based ephemeris.

Crater Detection. The crater detection method used in this work extends the implementation of the Mask R-CNN model presented previously.^{4,6} Based on training samples generated from the LROC global maps, the detector is able to handle images with a variable number of craters

in the field of view. While the detector model was built using PyTorch,⁷ the conversion to an Open Neural Network Exchange (ONNX) model interface⁸ enables integrated tests with the C++ implementation of the CNT workflow. Through the tools built for model conversion between PyTorch and ONNX, the detector model can be configured to handle different sized images which enables the use of other datasets for CNT testing.

Local Crater Catalog Generation and Identification. In order for detections to be used within the navigation filter their approximate locations on the Moon surface must be known, which is provided by the crater catalog.⁵ The catalog is initially truncated based on crater size, which is determined by the smallest expected crater that the satellite will be able to distinguish. This is based on the maximum/minimum orbit altitude and the camera resolution. Given the initial truncated database, a local catalog is calculated at each time in which measurements are received based on the field of view (FOV) of the spacecraft. The identification is then done through the Global Nearest Neighbor (GNN) approach via the Hungarian Matching algorithm. This approach finds the solution to

$$\hat{\pi} = \arg \min_{\pi \in \Pi} \left[\sum_{i=1}^n \min(c, C(i, \pi(i))) \right], \quad (1)$$

where c is the cutoff value for assignment, which is chosen by the user, Π is the set of all assignments,

$$C(i, j) = \|\mathbf{d}(i, j)\|_2 = \|\mathbf{r}_{detected}(i) - \mathbf{r}_{catalog}(j)\|_2, \quad (2)$$

$i = 1 : n, j = 1 : m, \mathbf{r}_{detected}$ is the position of a detected crater and $\mathbf{r}_{catalog}$ is the position of a crater in the local catalog. The addition of a bias on the measurements leaves the algorithm susceptible to false identification of craters. In order to combat this, two reasonable precautionary steps are taken before performing the identification. First, the Euclidean distance is taken between all measured craters, and if two craters are within a selected cutoff distance, then one of them is discarded. Additionally, the estimated bias output from the filter is subtracted from all the detections before matching is done. The alteration of the detection is only done within the matching algorithm. The detections which are passed into the filter are unaltered.

Measurement Model. The camera measurements are defined in the pixel space such that

$$\mathbf{h}(x, y, z) = \begin{bmatrix} x \\ z + b_x \\ y \\ z + b_y \end{bmatrix}^T, \quad (3)$$

where x, y and z are components of the vector from the camera to the surface feature in the camera (CAM) frame and b_x and b_y are the measurement bias in pixel space. Figure 1 shows the camera view of the craters as well as the axes for the camera frame. The camera frame corresponds to the North, East, Down (NED) frame in the current framework but the CNT algorithm

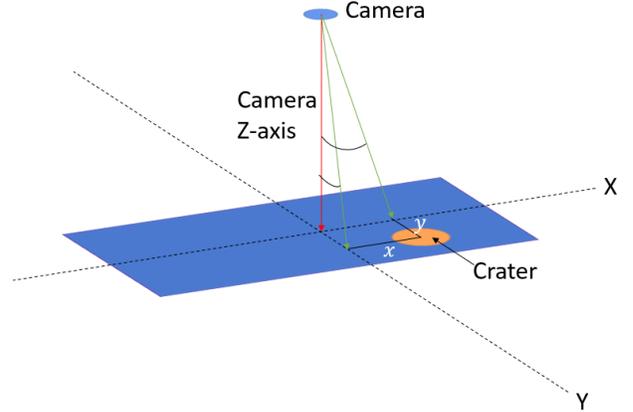


Figure 1. Camera Axes

allows for other attitude representations. The bias is currently assumed to be random and fixed but the model for the bias can easily be changed since the algorithm is implemented in an object-oriented framework. The partials in terms of the position of the surface feature are

$$\frac{\partial \mathbf{h}}{\partial [\mathbf{r}_{sf}^{CAM}]_{CAM}} = \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \end{bmatrix}, \quad (4)$$

where $[\mathbf{r}_{sf}^{CAM}]_{CAM}$ is the position of the surface feature in the camera frame. Finally, the partials are transformed into the Moon-Centered Inertial (MCI) frame,

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial [\mathbf{r}_{orb}]_{MCI}} &= \frac{\partial \mathbf{h}}{\partial [\mathbf{r}_{sf}^{CAM}]_{CAM}} \frac{\partial [\mathbf{r}_{sf}]_{MCI}}{\partial [\mathbf{r}_{orb}]_{MCI}} \\ &= \frac{\partial \mathbf{h}}{\partial [\mathbf{r}_{sf}^{CAM}]_{CAM}} * \mathbf{T}_{MCI}^{CAM}, \end{aligned} \quad (5)$$

where $[\mathbf{r}_{orb}]_{MCI}$ is the orbital position of the vehicle in the MCI frame and \mathbf{T}_{MCI}^{CAM} is the transformation matrix from the camera frame to MCI. The final measurement matrix is

$$\mathbf{H} = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial [\mathbf{r}_{orb}]_{MCI}} & \mathbf{0}_{2 \times 3} & \mathbf{I}_{2 \times 2} \end{bmatrix}. \quad (6)$$

Measurement Prediction and Update. The measurement update uses the EKF equations to update the state deviation resulting from each measurement at each time. All the measurements at a time are processed one at a time. Additionally, during each measurement update, measurement editing is performed to ensure that the measurements are statistically consistent with a chi-squared distribution with two degrees of freedom. The measurement editing is performed by calculating the squared Mahalanobis distance,

$$m_r^2 = \Delta \mathbf{z}_k^T (\mathbf{P}_k^T) \Delta \mathbf{z}_k, \quad (7)$$

where $\Delta \mathbf{z}_k$ is the measurement residual and \mathbf{P}_k is the innovations covariance.⁹ The measurements are discarded if their squared Mahalanobis distance exceeds the chi-squared value corresponding to 95%, which is 5.9915 for a distribution with two degrees of freedom. Once all measurements have been accounted for, the total deviation is added to the a priori state to obtain the a posteriori. The covariance is updated with each update of the deviation as well. The a posteriori covariance is obtained after the final measurement at that time has been processed.⁹

The prediction step is consistent with the process for a traditional EKF prediction. The CNT algorithm is capable of considering many different sources of perturbations in the propagation of both the state and the PNTM, these include gravity gradient perturbations, solar radiation perturbations and ephemeris-based location of the perturbing bodies. The PNTM is defined as

$$\mathbf{\Gamma}(t_k, t_{k+1}) = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau) \mathbf{B}(\tau) d\tau, \quad (8)$$

where $\mathbf{\Phi}$ is the state transition matrix (STM) and \mathbf{B} is a function of time relating the process noise to the state. The PNTM is initialized as matrix of zeros and after is approximated numerically.¹⁰

Time Bias Estimation. Time bias estimation is essential to correct errors in the onboard clock. The CNT algorithm uses ground-based ephemeris to correct on-board time and provide an updated a priori to the navigation algorithm. It is not realistic to assume that the satellite will receive a ground update at every measurement update, so for simulation purposes, a time update is done at a user-chosen frequency. The time bias is

$$\Delta t = t - t^*, \quad (9)$$

where t is the true time and t^* is the time measured by the clock on the spacecraft. The current CNT algorithm will estimate Δ given the following assumptions:

- There is no error in the time tag of the ground-based ephemeris.
- The clock bias is estimated using one navigation solution rather than a batch.
- The potential uncertainty in the ground solution is not considered.
- The bias estimator considers the reference state to be true.
- The navigation covariance does not change over the bias period when propagating.
- The algorithm does not consider the velocity covariance in the navigation solution.

- The algorithm assumes that:

$$\frac{\partial \mathbf{r}(t^* + \Delta t)}{\partial \Delta t} \approx \mathbf{v}(t^* + \Delta t) + (\mathbf{a}(t^* + \Delta t)) \Delta t. \quad (10)$$

In order to actually perform the bias estimation, the CNT software solves the nonlinear least squares problem

$$\widehat{\Delta t} = \arg \min_{\Delta t} (\mathbf{r}(t) - \hat{\mathbf{r}}(t^* + \Delta t))^T * P^{-1} (\mathbf{r}(t) - \hat{\mathbf{r}}(t^* + \Delta t)), \quad (11)$$

where a solution is converged upon when the Δt is found which minimizes the difference between the filter solution, $\hat{\mathbf{r}}(t^* + \Delta t)$, and the ground epoch solution, $\mathbf{r}(t)$.⁴

Image Generation and Processing. Testing the integrated performance of the position, navigation, and timing (PNT) methods developed in this work requires the generation of realistic imagery that is representative of lunar terrain. Previous work⁶ developed the capability to generate images from the LROC global map images,¹¹ which also enabled the generation of training data for the chosen crater detector used in this work. However, the use of the same image generation tools later led to large state errors observed in integrated tests that were previously characterized as some unknown systematic measurement bias.⁴ The previous errors have since been identified to be caused by inaccurate image projection and camera model assumptions that result from the use of the image generation tools used for detector training. To further study these issues, analysis of the image distortion are presented along with a new image generation method using crater templates are discussed below.

Distortion Analysis. The ML-based detector used in this work relies on an image generation process that produces images of a fixed image size, which takes the large sections of the LROC global maps to generate image samples of the specified crop size. As they were used for training the neural network, the cropped images were selectively sampled according to the training/testing paradigm described in previous work⁶ such that the detector can “learn” characteristics about lunar craters to enable detection. The focus of the image generation pipeline was not to render camera-specific terrain images from simulated orbits, instead, the tools’ primary aim was to create thousands of crater samples for training to develop a detector that was agnostic to specific orbital geometry, lighting conditions, and camera models. Along with these cropped images, the training datasets also include the catalogued crater data⁵ projected into a flat plane. The projection of the crater centroids to the flat cropped images and their expected crater locations from an orbital perspective do not match exactly, since the generated images do not account for the orthographic projection of the data onto the Moon’s surface, and the camera model used was an approximation of a pinhole model based on the cropped image size and a chosen altitude. The previous configuration of

the image generation tool to produce images from a simulated trajectory did not accurately account for these projection errors, which led to an observed error shown in Fig. 2. This plot shows the difference between the crater centroids in the expected locations based on an orbital camera model and those from the image generation truth data. For the 4000 craters included in this test, the error increases radially from the center of the image.

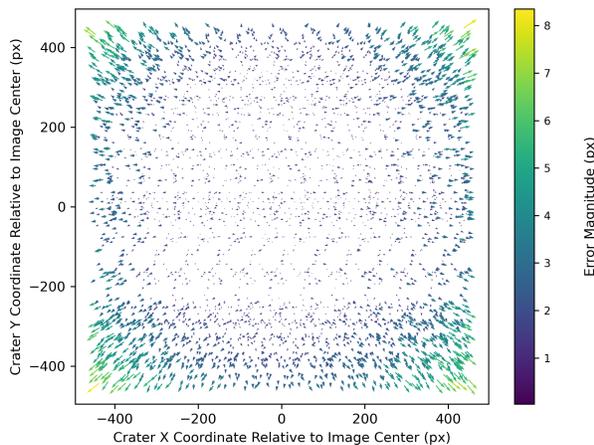


Figure 2. Radial pixel error between expected crater locations and image crater centroids before projection correction.

The 8 pixel error observed in the image corners corresponds to approximately 800 meters in crater centroid error, based on the current camera model used with a spacecraft simulated at 100 kilometers in altitude. The radial pattern suggests that the error can be approximated through the use of the Brown distortion model¹² (neglecting tangential distortion parameters)

$$\begin{aligned} x_f &= x + (x - x_c)(1 + k_1r^2 + k_2r^4 + k_3r^6), \\ y_f &= y + (y - y_c)(1 + k_1r^2 + k_2r^4 + k_3r^6), \end{aligned} \quad (12)$$

where (x, y) are the expected crater centroid coordinates; $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ is the centroid's distance from the (x_c, y_c) image center; (x_f, y_f) are the flat image centroid coordinates; and k_1, k_2, k_3 are the barrel distortion coefficients. Given the data shown in Fig. 2, a least squares solver is used to provide estimates of the distortion coefficients in Table 1.

Table 1. Distortion Parameters

Parameter	Value
k_1	2.838e(-8)
k_2	1.141e(-14)
k_3	-2.281e(-20)

Figure 3 shows the error magnitude as a function of the centroid distance in pixels. From the computed distortion coefficients, the error is approximated using the Brown

model shown in Eq. 12, which follows the trend of the raw data. However, the output of the Brown model does not capture the behavior of the non-zero error close to the image center of approximately 1 pixel in magnitude, which will contribute to the measurement error resulting from centroid estimates that use this distortion map.

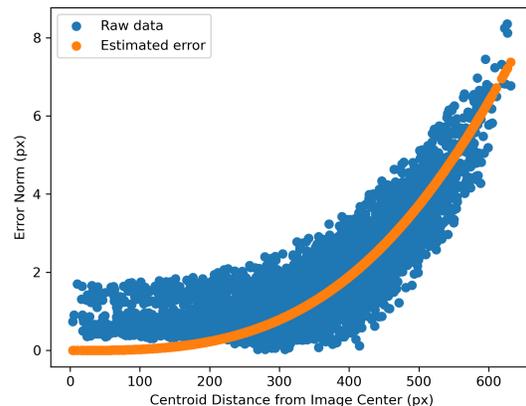


Figure 3. Error magnitude compared to the Brown model output.

Because of this identified distortion, the centroid measurements can be constrained to exclude values found in the outer corners of the image, at the cost of limiting the number of craters available for subsequent processing within CNT. User-configurable flags can be also be used to ignore craters that are partially in the field of view such that the crater detector and the subsequent ellipse fitting method are able to compute more accurate centroids given full craters. Limiting the detections to exclude the far corners of the image and the edges reduces the error of the computed centroids with the goal of preventing inaccurate matching between the detections and the catalog in the crater identification step.

Template Images. An alternate image generation capability based on creating image templates was developed for testing in this work. Since the detector was developed using the aforementioned image generation tools, the same software can be reconfigured to produce image samples of every known crater included in the Robbins catalog. Additionally, previous testing of the crater identification and position estimator components of CNT relied on simulated detections that were generated by sampling and propagating expected crater locations in the satellite's field of view for a given trajectory. With crater templates generated for the catalog and knowledge of the expected craters in the field of view, image stitching methods are implemented to resolve the issues previously encountered with camera and projection errors without the need for radially distorting the images. An example generated image is shown in Fig. 4, where the chosen background terrain is sparsely populated with small

craters in a relatively flat environment. In this image, the larger craters are placed on top of the background terrain at their expected locations. One advantage of using this image generation method is the reduced runtime, as the templates can be generated prior to testing a particular trajectory which significantly reduces the image generation time compared to the previous method as it required file queries of large input maps. To efficiently run Monte Carlo tests for the CNT workflow, the template-based image generation method was chosen as the primary tool for the remaining work.

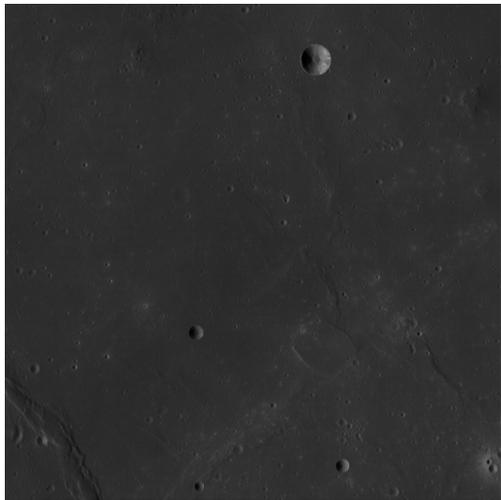


Figure 4. Template-based simulated image.

Test Data Generation and Test Cases. As described earlier, template-based images are generated using simulated detections. Simulated detections are generated by projecting the field of view (FOV), at a given point in time, onto the moon and retrieving the craters that fall within its vicinity. For this work, the camera model follows a simple pinhole model and the FOV is assumed to be square with 909 pixels on each side. Once the detections are retrieved, measurement noise is added. The angle between the sun, moon and the spacecraft is also tracked, and when the angle is greater than 90 degrees, no detections are generated. The shadow due to this angle is not accounted for yet. It is simply assumed that if the angle is less than 90 degrees consistent lighting is available and if the angle is greater than 90 degrees, the image is completely dark. The measurement bias is assumed to be random and fixed, so in simulation this is done by choosing a mean and covariance for the bias and sampling it for each MC trial. The bias is then added to each simulated detection in both the x and y camera directions. This work also tests trajectories with higher altitudes than the previous work,⁴ so the bias is scaled by a factor of $100/h_0$, where h_0 is the initial altitude, as the altitude increases. This step was taken because the initial bias was designed to push the limits of the algorithm at 100 km altitude but the same bias applied at a higher

altitude will equate to a larger projection of that bias onto the Moon. In practice, the bias will not change with altitude but the bias that is used for testing is also larger than should be expected in practice.

The simulated detections can be generated for any orbit parameters. For the purposes of this work, two 100 sample Monte Carlo tests are provided with images in-the-loop and nine 100 sample Monte Carlo tests are provided for simulated detections only. The tests with images take significantly longer to run than the tests with simulated detections, so using simulated detections allows for the assessment of the identification, state estimation and time bias estimation without taking the time of running images through the detector. Optimization of the detector could improve the evaluation time it takes to run images and is currently being performed. For all of these tests, the initial eccentricity (e), right ascension of the ascending node (Ω), argument of periape (ω), and true anomaly (ν) remain the same while semimajor axis (a) and inclination (i) are varied. The static variables are all set to zero except for eccentricity, which is set to a value of 0.003. For the two tests with images, $a = 1837.5$ km and 2337.5 km and $i = 10^\circ$. The simulated detections test cases consider $a = 1837.5$ km, 2037.5 km and 2337.5 km and $i = 20^\circ$, 50° , and 60° . For all test cases 20×20 gravity perturbations are included and DE430¹³ is used for propagation of the truth and DE421¹⁴ is used for propagation within the filter. The gravity perturbations are taken from a gravity clone generated from the Lunar Reconnaissance Orbiter (LRO) gravity data.¹⁵ The time bias for all tests is set to two seconds. The initial reference time January 1, 2000 at midnight. Additionally all tests initially start on the dark side of the orbit but they return to the light side after only a few minutes.

Results. Figures 5 and 6 show the state and bias errors, respectively, with an initial state of 10° inclination, semimajor axis of 1837.5 km, and a bias in both x and y is sampled from a mean of 2×10^{-3} and standard deviation of 5×10^{-5} . Figures 7 and 8 show the state and bias errors, respectively, with an initial state of 50° inclination, semimajor axis of 2037.5 km, and a bias in both x and y is sampled from a mean of 6.67×10^{-4} and standard deviation of 1.667×10^{-5} . The bias mean and standard deviation are normalized by focal length, which is 982 pixels for the purposes of this paper, for ease of implementation. The 3D RMSE, which is given by

$$RMSE_{3D} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\delta x^2(t_i) + \delta y^2(t_i) + \delta z^2(t_i))}, \quad (13)$$

where δx , δy and δz are the state errors and N is the total number of measurement times, is shown in Table 2 along with the 3D RMSE for test cases with simulated detections.

Figures 5 and 7 indicates that the filter is pessimistic during periods of full shadow since the filter three sigma

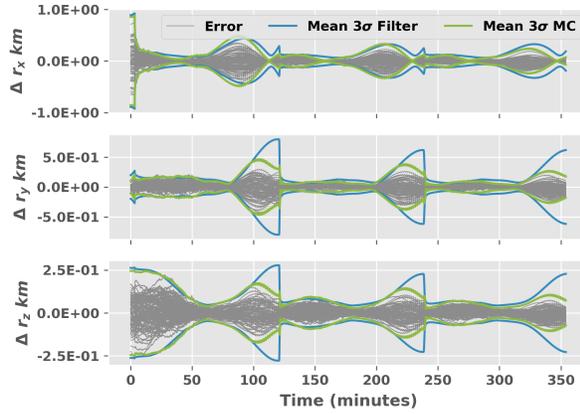


Figure 5. Monte Carlo Test Position Error with Simulated Detections, $i = 20^\circ$, $a = 1837.5$ km

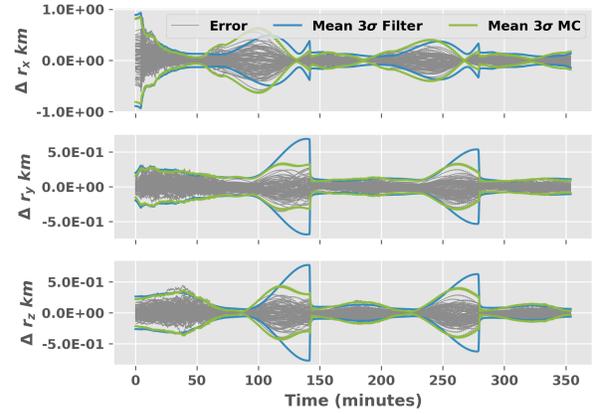


Figure 7. Monte Carlo Test Position Error with Simulated Detections, $i = 50^\circ$, $a = 2037.5$ km

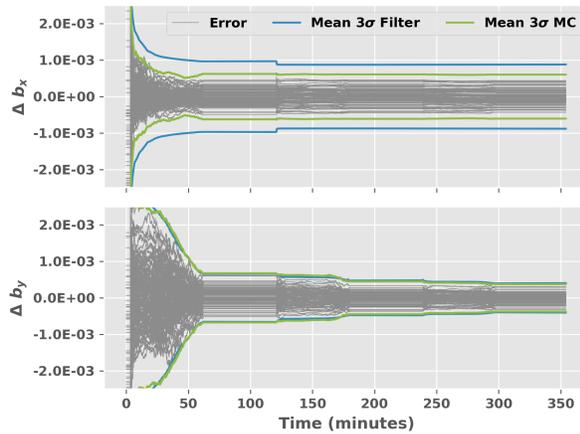


Figure 6. Monte Carlo Test Measurement Bias Error with Simulated Detections, $i = 20^\circ$, $a = 1837.5$ km

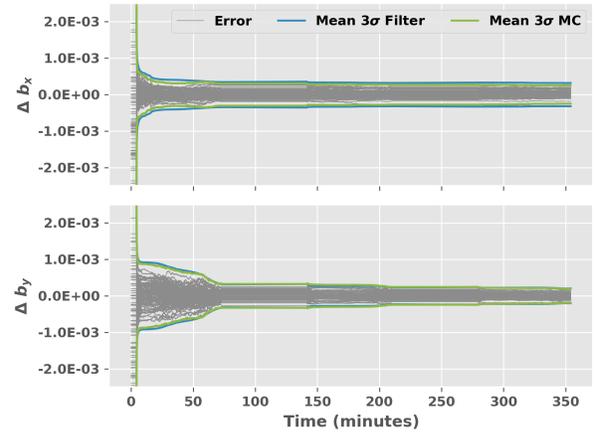


Figure 8. Monte Carlo Test Measurement Bias Error with Simulated Detections, $i = 50^\circ$, $a = 2037.5$ km

bound is larger than the MC three sigma bound. The filter and MC three sigma bounds are significantly closer during times where measurements are available though. The pessimism of the filter is most likely due to the nonlinearity of the problem. This could be improved by tuning of the process noise or finding a better model for the process noise. Additionally, the performance of the filter seems to degrade as inclination and/or altitude increases, which can be expected due to increased nonlinearity. Further work needs to be completed to discern whether the goal of a position error RMS less than 100 m in each axis can be met for these conditions. Currently only the tests at 60° inclination and the highest altitude case at 50° inclination violate this requirement. Despite the large covariance though during darkness, for all cases, the filter showed better consistency once measurements were available. There are also a few points in the x position error where the error is very small but the covariance bounds of the filter do not reflect this. This phenomenon is most likely due to the geometry of

the problem because of its repetitive nature.

Figure 9 shows the MC results when processing images for an inclination of 10° , semimajor axis of 1837.5 km, and no bias. The bias results in Figure 10 indicate that despite not adding any bias to the images, there is one. Figure 11 shows the bias results when adding the mean of all of the estimates at the final time to all of the data points. These results are clearly bounded and the MC and filter three sigma bounds are close together. This indicates that the filter is most likely estimating the bias correctly, but either the image generation process or the detector is adding a bias to the detections. While the exact cause of this bias is unknown, it will be investigated in the future. The position error displays that the bias does not significantly effect the results of the position error estimates. Figures 12 and 13 show the MC state error and adjusted bias error, respectively, with an inclination of 10° , semimajor axis of 2337.5 km, and no bias. The RMS errors for both tests with images are shown in Table 3 where 1 refers to the lower semimajor axis test and 2 refers

a (km)/ i (°)	20	50	60
1837.5	96.40	113.18	115.60
2037.5	117.68	129.89	164.08
2337.5	125.81	146.82	200.51

Table 2. Mean 3D RMS Errors for All Simulated Detections MC Tests

to the higher one. The RMS errors for the higher altitude test are slightly higher, which is consistent with simulated detections results. Both cases meet the requirement of less than 100 m error in each axis.

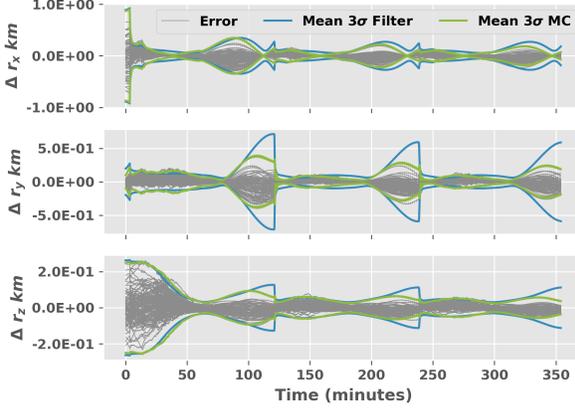


Figure 9. Monte Carlo Test Position Error, $i = 10^\circ$, $a = 1837.5$ km

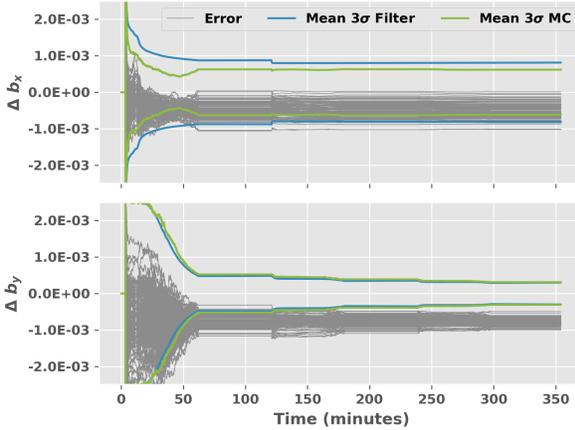


Figure 10. Monte Carlo Test Measurement Bias Error, $i = 10^\circ$, $a = 1837.5$ km

Future Work. The CNT algorithm is on track to meet its goals with simulated imagery but in order to push the algorithm to a higher technology readiness level, imagery from a lunar mission must be processed. Figures 14 and 15 show the detector output overlaid onto two images taken from the LROC narrow angle camera (NAC). The images have been cropped to be square and their overall dimensions are 316x316 pixels. The figures

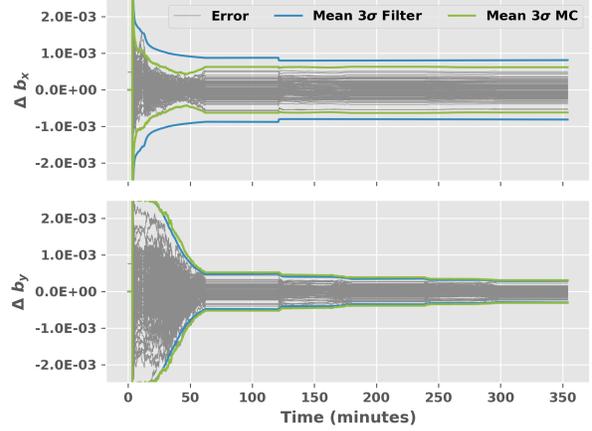


Figure 11. Monte Carlo Test Measurement Bias Error Adjusted by Mean of Final Points, $i = 10^\circ$, $a = 1837.5$ km

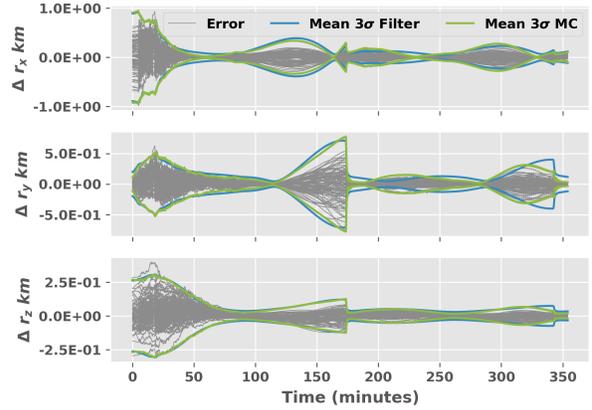


Figure 12. Monte Carlo Test Position Error, $i = 10^\circ$, $a = 2337.5$ km

evidence that the detector is successfully able to identify craters within on-board imagery. The future work is to retrieve more LROC imagery and trajectory information and run a set of imagery through the full CNT algorithm.

Additionally, the time bias and measurement bias models both need more robust implementations, such as random walk for the clock bias and a fixed, radial bias for the measurement bias at all altitudes. The CNT software must also be implemented on small satellite compatible hardware such as the NVIDIA Jetson. To further improve the testing pipeline, future work towards the generation of lunar images through the use of Blender and improved camera models will enable additional capabilities for CNT. Initial work is underway to generate images from Blender, such that the image projection issues encountered using the LROC global maps are mitigated. This addition to the image generation capabilities can also enable the inclusion of varying the simulated spacecraft's attitude and further testing the detector's sensitivity to

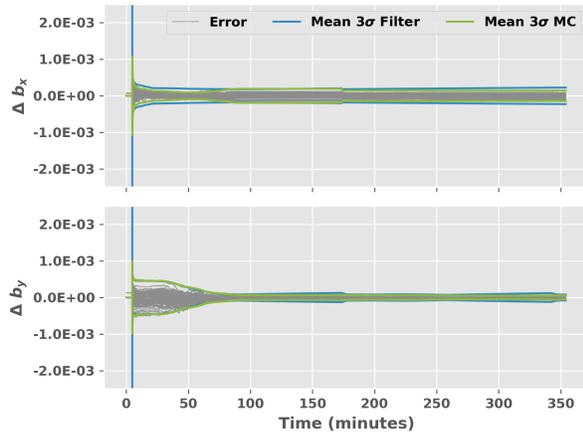


Figure 13. Monte Carlo Test Measurement Bias Error Adjusted by Mean of Final Points, $i = 10^\circ$, $a = 2337.5 \text{ km}$

	r_x	r_y	r_z	3D
mean 1 (m)	59.12	60.49	29.51	91.21
std 1 (m)	21.85	23.23	11.57	29.29
mean 2 (m)	86.75	82.34	35.41	127.19
std 2 (m)	29.68	28.54	16.70	36.82

Table 3. Mean and Standard Deviation of RMS Errors for MC Tests with Images

varying lighting conditions.

Conclusions. This work evidenced that the CNT algorithm is capable of meeting its goals of a position error RMS of less than 100 m and a clock estimate of less than 100 ms for the majority of tests ran. Further work will need to be accomplished to discover whether these goals can be met at up to a 500 km semimajor axis at high inclinations. It is also shown that the template-based imagery provides an opportunity to test the fully integrated algorithm without the issues provided by the mosaic images. The results with the template images and the preliminary results with the LROC NAC camera cropped images are good indications that the CNT algorithm is capable of performing on-board navigation and timing estimates. Future work will provide full results with LROC NAC images in-the-loop.

References.

- [1] F. C. Hanak, *Lost in low lunar orbit crater pattern detection and identification*. PhD thesis, The University of Texas at Austin, 110 Inner Campus Drive, MAI 101, Austin, TX 78712, 5 2009.
- [2] L. M. Downes, T. J. Steiner, and J. P. How, “Neural network approach to crater detection for lunar terrain relative navigation,” *Journal of Aerospace Information Systems*, pp. 1–13, 2021.
- [3] E. Emami, G. Bebis, A. Nefian, and T. Fong, “Automatic crater detection using convex grouping and convolutional neural networks,” pp. 213–224, 12 2015.
- [4] Z. R. McLaughlin, R. E. Gold, S. G. Catalan, R. Moghe, B. A. Jones, and R. Zanetti, “Crater navigation and timing for autonomous lunar orbital operations in small

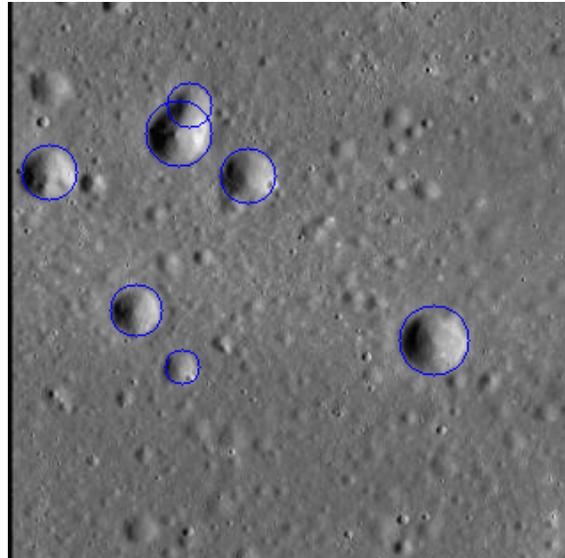


Figure 14. LROC NAC Image 1 with Detector Output

satellites,” in *Rocky Mountain AAS GN&C Conference*, February 2022.

- [5] S. J. Robbins, “A new global database of lunar impact craters >1–2 km: 1. crater locations and sizes, comparisons with published databases, and global analysis,” *Journal of Geophysical Research: Planets*, vol. 124, no. 4, pp. 871–892, 2019.
- [6] S. G. Catalan, J. S. McCabe, and B. A. Jones, “Implementation of machine learning methods for crater-based navigation,” in *AAS/AIAA Astrodynamics Specialist Conference*, no. AAS 21-519, (Big Sky, MT Virtual), August 2021.
- [7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.
- [8] J. Bai, F. Lu, K. Zhang, *et al.*, “Onnx: Open neural network exchange.” <https://github.com/onnx/onnx>, 2019.
- [9] R. J. Carpenter and C. N. D’Souza, “Navigation filter best practices,” Tech. Rep. TP–2018–219822, National Aeronautics and Space Administration (NASA), NASA Engineering and Safety Center, Hampton, Virginia, April 2018.
- [10] B. D. Tapley, B. E. Schutz, and G. H. Born, *Statistical Orbit Determination*. Elsevier Academic Press, 1st ed., 2004.
- [11] E. Speyerer, M. Robinson, B. Denevi, *et al.*, “Lunar reconnaissance orbiter camera global morphological map of the moon,” in *Lunar and Planetary Science Conference*, no. 1608, p. 2387, 2011.
- [12] D. C. Brown, “Close-range camera calibration,” *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.
- [13] W. Folkner, J. Williams, D. Boggs, R. Park, and P. Kuchynka, “The planetary and lunar ephemerides de430 and de431,” *Interplanetary Network Progress Report*, vol. 42–196, February 2014.

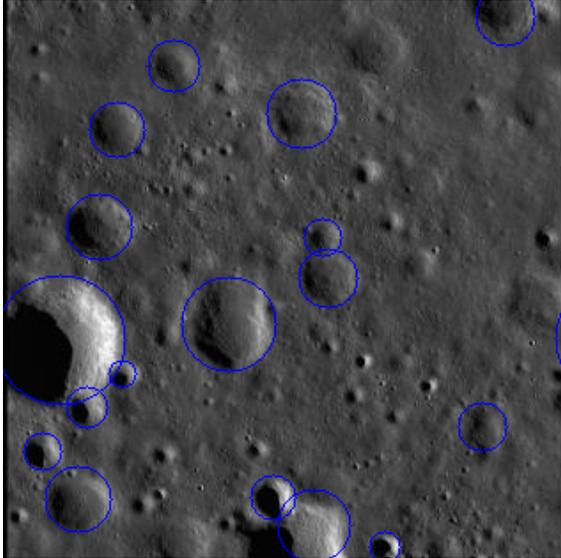


Figure 15. *LROC NAC Image 2 with Detector Output*

- [14] W. Folkner, J. Williams, and D. Boggs, “The planetary and lunar ephemeris de 421,” *Interplanetary Network Progress Report*, vol. 42–178, August 2009.
- [15] T. J. Shabaka, J. B. Nicholas, S. Goossens, F. G. Lemoine, and E. Mazarico, “Error propagation for high-degree gravity models developed from the grail mission,” *PDS-geosciences*, April 2014.