AAS 22-113

VISUAL ODOMETRY FOR PRECISION LUNAR LANDING

Giovanni Molina[*], Michael Hansen[†], Joel Getchius[‡], Randall Christensen[§], John A. Christian[¶], Shaun Stewart[‖], and Tim Crain[**]

Intuitive Machines has developed a state-of-the-art precision landing and hazard avoidance (PLHA) system that will enable its Nova-C lunar lander to safely touch down on the lunar surface. This system consists of an array of sensors that includes an inertial measurement unit (IMU), optical camera, and a laser range finder sensor (LRFS) as well a series of algorithms which process and fuse the sensor data and produce measurements used in the navigation system. One of the measurements utilizes image processing and visual odometry (VO) to compute a delta-position (DPOS) measurement which describes the lander's direction of motion between two image captures. In this paper, we detail the development and implementation of this measurement for the Nova-C lander, demonstrate our rigorous testing methodologies and present our findings and results.

INTRODUCTION

For the first time since the Apollo 17 landing in 1972, Intuitive Machines will return The United States of America to the lunar surface. The Nova-C lander will deliver several NASA payloads to the lunar surface for various scientific and engineering research purposes. The structural requirements for the Nova-C lander require that the navigation system determines the lateral velocity to within 0.1 m/s at touchdown. This requirement is met via the PLHA system, which utilizes two star trackers, two IMUs, a Terrain Relative Navigation (TRN) and a Hazard Relative Navigation (HRN) optical cameras, each co-boresighted with a laser range finder.

[*]Software Developer, Intuitive Machines, 3700 Bay Area Blvd, Houston, TX 77058.
[†]Nova-C Navigation Lead, Intuitive Machines, 3700 Bay Area Blvd, Houston, TX 77058.
[‡]Sr. Navigation Systems Engineer, Omitron Inc., 7051 Muirkirk Meadows Drive, Suite A, Beltsville, MD 20705.
[§]Assistant Professor, Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322.
[¶]Associate Professor, Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332.
[‖]Nova-C Flight Dynamics Lead, Intuitive Machines, 3700 Bay Area Blvd, Houston, TX 77058.
[**]Chief Technology Officer, Intuitive Machines, 3700 Bay Area Blvd, Houston, TX 77058.

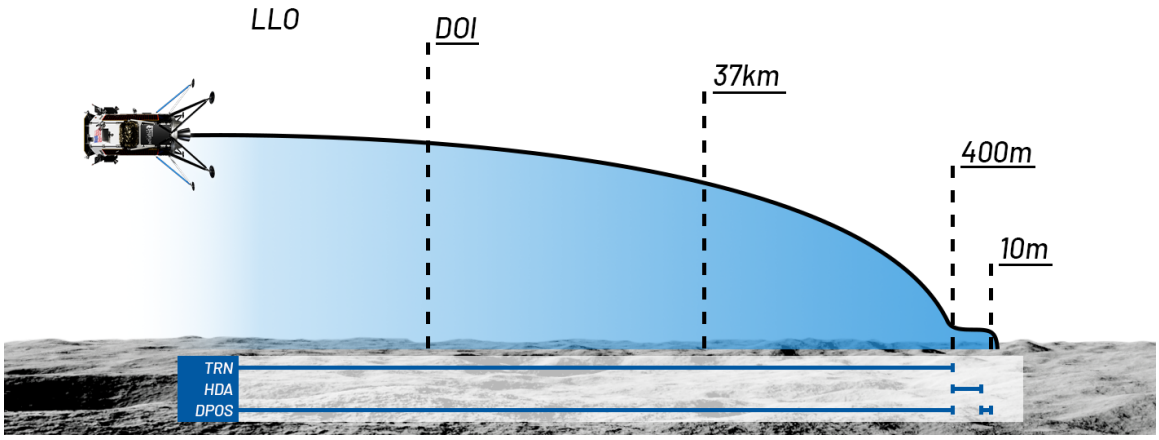


Figure 1. IM-1 mission trajectory overview showing the active optical navigation algorithms from LLO to Touchdown

Figure 1 roughly outlines the operational timeline for the use of the PLHA system. In low lunar orbit (LLO), the DPOS measurement is constructed with feature tracking from two images taken in successive times of the lunar surface. Crater identification and tracking is also used provide line of sight measurements. Both these measurements are primarily formulated with the TRN sensor suite, and performance is verified against ground based radiometric tracking and orbit determination.

After 24 hours in LLO, the Nova-C performs its de-orbit initialization burn (DOI), which begins to lower the altitude of the vehicle. After DOI, the lander coasts to the powered descent initialization (PDI) while rotating such that the TRN camera and LRFS point as close to nadir as possible, but ensuring the engine bell is pointed in the velocity direction for PDI. We note that the HRN sensor suite is pointed to the sky and is of little utility. During this coast, crater tracking and DPOS measurements are once again utilized in the on-board Kalman filter, and LRFS measurements become available at an altitude of about 65 km. At an hour post DOI, PDI begins and the vehicle constantly is thrusting until touchdown. No change in the concept of operations for sensors occurs at PDI.

Once the lander is at an altitude of 400 meters and is 400 meters downrange from the intended landing site, it pitches over such that the HRN sensor suite begins to observe the intended landing site. Custom hazard detection and avoidance (HDA) algorithms are then executed and, if necessary, an alternative landing site is selected.

Post HDA, the lander PLHA system relies only on DPOS and LRFS measurements to correct integrated errors from the system IMU. While LRFS is expected to provide robust observability in altitude and altitude rate, it has no observability in the lateral velocity. Therefore, DPOS is necessary to arrest the errors in the lateral velocity estimate and maintain the landing condition requirements on the PLHA system.

In this paper, we briefly review the mechanism for formulating the DPOS measurement, discuss its implementation into the PLHA Kalman filter, and review the results from various test campaigns.

VISUAL ODOMETRY AND THE DELTA-POSITION (DPOS) MEASUREMENT

Visual odometry is an established computer vision technique[1–3] that estimates the change-in-pose between two images up to an unknown scale. This task is often accomplished by tracking the apparent motion of common features in an a sequence of two images. These features may be opportunistically detected (e.g., with a feature descriptor) and there is no need to ever know (or solve for) the position of these features. This is in contrast to simultaneous localization and mapping (SLAM) techniques, which require explicit modeling and estimation of the world coordinates of observed features.

In many spacecraft navigation applications there is no need for an image-based TRN system to explicitly estimate full pose from image data alone. This is particularly the case when excellent attitude knowledge is available from star trackers, IMUs, and other attitude sensors. This is expected to be the case for the Nova-C lunar lander. In this situation, the visual odometry solution may be considerably simplified to produce a direction-of-motion measurement.[4]

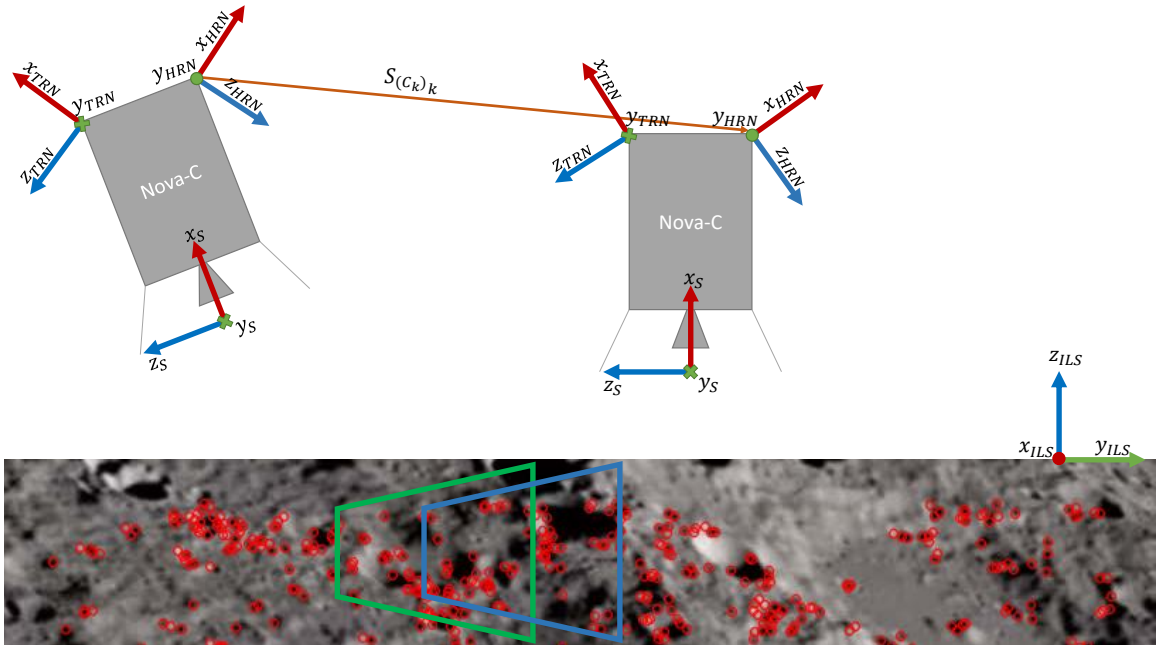


Figure 2. Illustration of the process of computing a single DPOS measurement. The DPOS measurement is the unit vector that describes the direction of motion leftmost spacecraft HRN camera position to the rightmost spacecraft HRN camera position expressed in the rightmost HRN camera frame. The green and blue trapezoids represent the projected field of views of the two image captures.

As part of the Nova-C PLHA system, we've adapted the direction-of-motion measurement from Ref. 4, which we refer to as a DPOS measurement within the flight software application. As depicted in Figure1, DPOS measurements will be available through most of the mission timeline starting at LLO and continuing throughout until 10m above the ILS surface. DPOS measurement processing is disabled at 10m to avoid incorrect measurements due to debris from the engine exhaust plume moving along the cameras field of view. Linear covariance analysis has shown that doing this is safe and that we can achieve a soft landing. For the Nova-C navigation system, DPOS is required

to provide a measurement once per second.

To produce a DPOS measurement we require two images taken some time apart and the change in camera orientation from one image capture time to the next. As illustrated in Figure 2, the measurement produced by DPOS is the unit vector $S_{(c_k)_k}$ that gives the three-dimensional direction-of-motion from the camera position at time *k-1* to the camera position at time k, expressed in the second camera frame $(C_k)_k$.

DPOS FLIGHT SOFTWARE IMPLEMENTATION

We developed DPOS as a fully parameterized software application, allowing us to control its operation parameters via pre-populated configuration tables as well as giving us the option to change the parameters on the fly. In this section we discuss our work to transform the DPOS measurement algorithm from theory into a robust flight software application, ready for a realtime navigation system. First, we discuss sensor and other hardware requirements for accurate measurements. Then, we describe the different pieces of software implemented ranging from managing the input images, to the computer vision and image processing aspects.

Sensor Requirements

As any optical navigation algorithm, DPOS requires the use of high quality, high resolution cameras as the main source of data fed into the processing pipeline. We equipped the Nova-C lander with two cameras on opposing ends of the lander structure. One of the cameras will serve for the main purpose of TRN while the other camera serves as the HRN camera. DPOS is able to access both cameras one at a time, and which camera is in use is selected by the Visual Processing Flight Manager (VFM). The VFM acts as an intermediary controller between the Autonomous Flight Manager (AFM) and the Optical Navigation (OpNav) applications and simplifies the operation of the cameras.

In terms of optical characteristics for the cameras we have ensured that the field of view is wide enough to provide sufficient observable area for feature tracking as well as focusing to infinity to have an optimal in-focus working distance. Our camera system is able to control the exposure times for the fixed aperture camera to capture images with great contrast and low motion blur. Finally, the camera resolution is high enough to be able to resolve surface features on the Moon from all altitudes. For the OpNav software, we require 8-bit images from the cameras, with a minimum transfer rate of 1 Hz.

Other than cameras, DPOS makes indirect use of the IMUs, star trackers and other sensors that provide altitude information by accessing the navigation system's attitude estimates. DPOS runs on a semi-dedicated Visual Processing Unit (VPU), a 1 GHz single-core rad hard processor that runs on the Buildroot Linux OS.[5] All of these sensors are located in two purpose-built chassis dubbed the Navigation Pods or Navpod.

Feature Detection

In the field of computer vision and image processing, feature detection and feature matching have been widely used for the analysis of motion in image sequences. A feature in this context is any part of an image that is *interesting* or *salient* when compared to the rest of the image. The saliency of such features is determined by algorithms classified as *feature detectors* which use computer vision concepts such as finding edges and corners in an image to generate the features. Once these

features are detected, another algorithm called the feature descriptor builds a representation of the feature based on the center pixel of the feature along with a specific sized window around it. This descriptor is usually a numerical vector of varying sizes which allows a simple comparison between descriptors in a given vector space. The end result after feature detection and description is a pair of the pixel coordinates of the feature center in the image frame, named a keypoint, and its descriptor. Figure 3 showcases an image with the detected features overlaid over it, and a zoomed in feature showing the window used to build its descriptor.

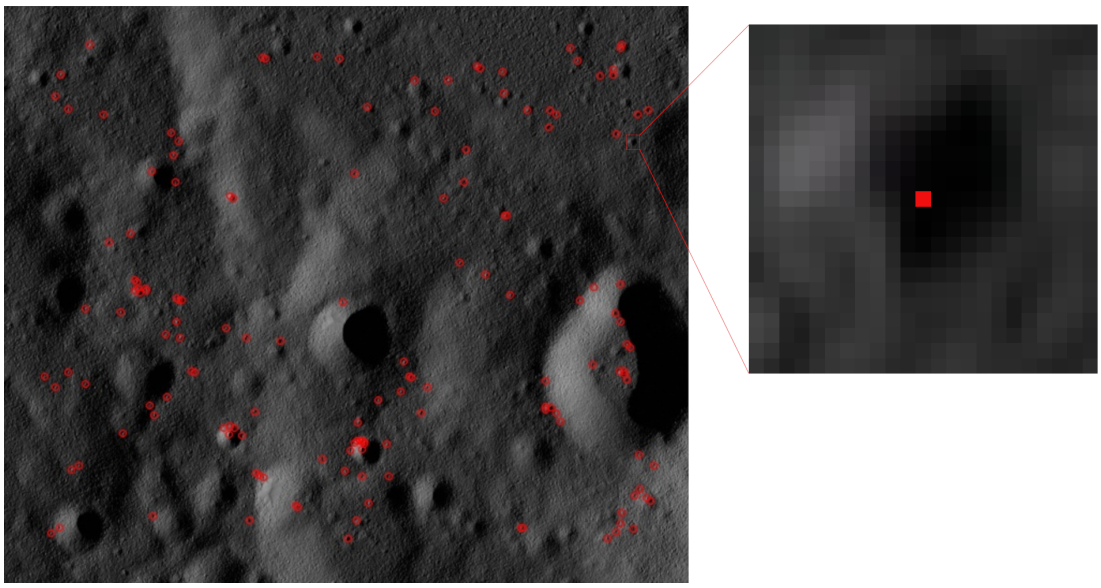
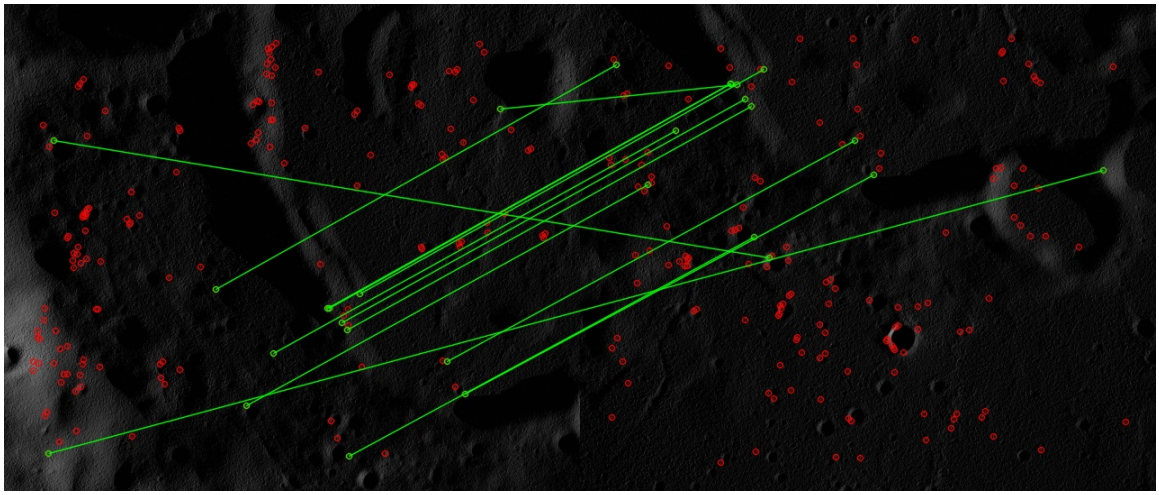


Figure 3. On the left we have a synthetic images with feature locations plotted on top, and on the right we zoom into one of the features to show what a descriptor patch looks like.

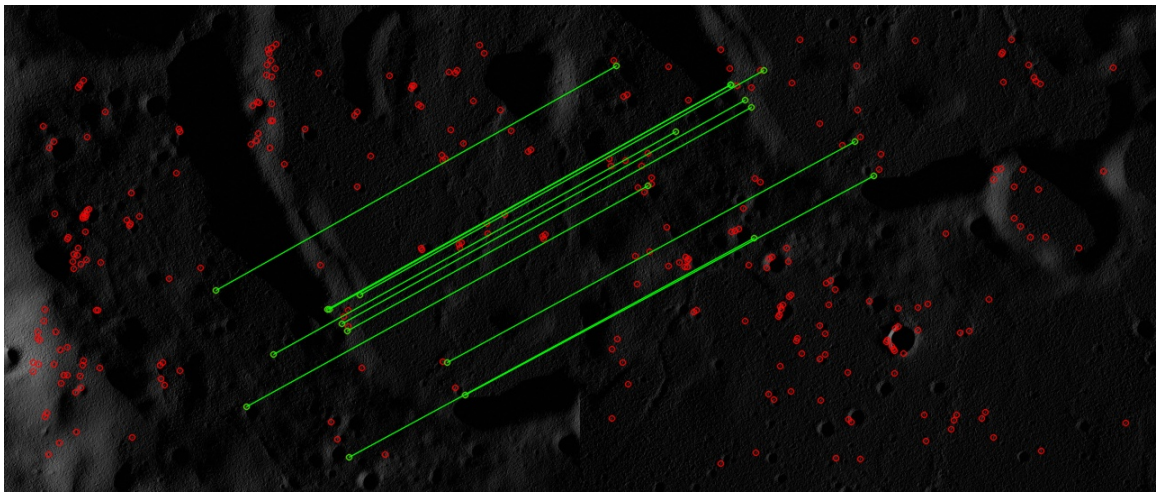
Multiple feature detectors exist today, such as AKAZE,[6] FAST,[7] SIFT,[8] SURF,[9] and ORB.[10] For DPOS, we selected the ORB feature detector and descriptor since these are known to be both robust and efficient to compute.[11–18] Moreover, our own internal analysis and testing indicated that ORB has good repeatability, scale and rotation invariance, and stability in the presence of image noise. Following the ThinVPU[19] concept, we leverage the open source software library OpenCV[20] in the implementation of the image processing pipeline for DPOS.

Feature Matching and Outlier Rejection

The next stage in the image processing pipeline involves identifying features that are common to both images utilized by DPOS. This process, known as feature matching, utilizes a matching strategy that gives an abundant amount of feature matches while ensuring that the matching points are correctly identified. In our implementation, outlier rejection is also a critically important aspect of the matching strategy. An outlier is a pair of points that were erroneously matched. This can happen if the feature descriptors are similar and no better match exists.



(a) Outliers



(b) Inliers

Figure 4. Feature matches with outliers before RANSAC and feature matches with only inliers after RANSAC.

In our matching strategy, we reject outliers in two ways. First we find the matches using a brute-force matcher that returns the two closest matching points from the second image for each point in the first image. We then apply Lowe's Ratio Test[8] to obtain our first set of inliers, or correctly matching points. A secondary outlier rejection algorithm is implemented using Random Sample Consensus (RANSAC).[21] In our RANSAC implementation, we compute the DPOS measurement with a minimal set of randomly selected matches from our current inliers and then compute the Sampson Distance for the remaining set of matches, much akin to the methodology in Refs. 4, 22. Any match that has a Sampson Distance below our acceptance threshold is classified as an outlier. We repeat this N number of times with a different random set of minimal inliers and keep the complete set of matches with the highest number of inliers. Once we have found our final set of inliers, we compute the DPOS measurement with the full set of inliers (instead of a minimal set of matches) to obtain our final result. Figure 4 shows a set of matches that contains 3 outliers in 4(a)

and the resulting set of inliers after RANSAC in 4(b).

Finally, we found that the feature detection and matching were the most computationally expensive parts of the algorithm during our performance testing and analysis. In order to maintain our required 1 Hz measurement output we down-sample the image by a factor of 0.25 before feature detection and matching. This linearly increases the computation speed 4x and allows us to run DPOS in under a second, while maintaining an acceptable measurement output with nearly no loss in information.

Image Selection

The DPOS measurement performance is highly correlated to the overlap between the two images. If too much overlap occurs, and the relative motion is not observed; with too little overlap, there will not be enough common features between the two images to realize an accurate measurement of the direction of motion. Therefore, the Nova-C DPOS implementation includes logic to estimate the time between images in order to achieve as close to 50% overlap between images as possible. Analysis of the ideal image overlap revealed that 50% image overlap minimizes the DPOS error.

A simplified, rectangular projection of the camera field of view on the lunar surface is utilized to compute how much time would elapse for 50% overlap given the navigation estimates of altitude, ground speed, and attitude. An example of the dynamic time between images is shown in Figure 5, where the time needed between images to achieve the 50% overlap decreases as the altitude decreases during transit. During terminal descent, this computation is bypassed in lieu of a pre-loaded delta time based on the pre-flight trajectory design. The switch is necessary due to the difference in geometry of the features' movement in the field of view of the cameras.

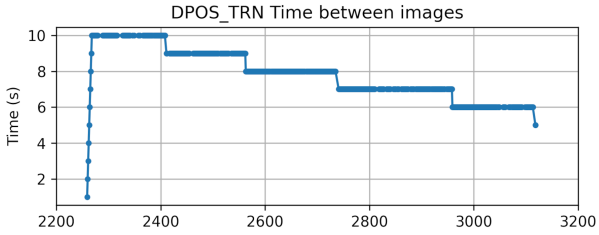


Figure 5. Plot showing the dynamic time needed between images to obtain a 50% image overlap.

DPOS NAVIGATION KALMAN FILTER IMPLEMENTATION

As previously mentioned, the Nova-C PLHA system sensor suite consists of star tracker measurements for attitude determination, two IMUs for acceleration and gyro measurements, two laser range finders, and two cameras that are used to produce measurements of known craters as well as DPOS observations. These measurements are fused with a 1 Hz hybrid Extended Kalman Filter / Linearized Kalman Filter (EKF/LKF), while the state space is advanced in time with a 50 Hz "fast propagation" routine. The specific 30 element state space contains:

- Position vector of the selected IMU in the inertial frame, $\vec{r}_I^{IMU/I}$
- Velocity vector of the selected IMU in the inertial frame, $\vec{v}_I^{IMU/I}$

- Gibbs vector of the change in the reference attitude in accordance to the Multiplicative Extended Kalman Filter formulation, $\vec{\Delta g}_{REF}^{REF}$
- Accelerometer bias vector, $\vec{b}_{accel}$
- Gyro bias vector, $\vec{b}_{gyro}$
- Laser range finder bias state (one for each sensor)
- 12 elements for the misalignment states of the two star trackers, HRN, and TRN pods
- A sub-spacecraft terrain bias

As DPOS is a measurement of direction of motion, the hybrid EKF/LKF models a DPOS measurement that utilizes a first image at time t_0 and second image at time t_1 per Eq. (1)

$$\vec{\rho}(t_1) = \frac{T_I^{cam_{t1}} \vec{r}_I^{IMU/I}(t_1) - T_I^{cam_{t0}} \vec{r}_I^{IMU/I}(t_0)}{\left| \vec{r}_I^{IMU/I}(t_1) - \vec{r}_I^{IMU/I}(t_0) \right|} \tag{1}$$

Where $T_{I_t}^{cam}$ is the transformation from the inertial frame to the camera frame at time t_1 / t_0. Of particular concern is the evaluation of $\vec{r}_I^{IMU/I}(t_0)$, which must be dynamically continuous with $\vec{r}_I^{IMU/I}(t_1)$ with the equations of motion and the IMU data. However, in lieu of buffering IMU data for the time necessary and complicating the flight software with another propagation routine, a buffer of the Kalman filter states is maintained. To ensure consistency between the states, the buffer contains the state transition matrix as well as the delta state at each Kalman filter cycle.

Setting $\vec{\alpha} = \vec{r}_I^{IMU/I}(t_1) - \vec{r}_I^{IMU/I}(t_0)$, the measurement can be approximated as a linear equation:

$$\begin{aligned} \delta\vec{\rho}(t_1) \quad = \quad & \frac{\delta\vec{\rho}}{\delta\vec{r}_I^{IMU/I}(t_1)} \delta\vec{r}_I^{IMU/I}(t_1) \quad + \\ & \frac{\delta\vec{\rho}}{\delta\vec{r}_I^{IMU/I}(t_0)} \delta\vec{r}_I^{IMU/I}(t_0) \quad + \\ & \frac{\delta\vec{\rho}}{\delta\vec{\Delta g}_{REF}^{REF}} \delta\vec{\Delta g}_{REF}^{REF} \quad + \\ & \frac{\delta\vec{\rho}}{\delta\vec{\theta}_{cam}} \delta\vec{\theta}_{cam} \end{aligned} \tag{2}$$

To convert the position at t_0 to the position at t_1 the state transition matrix can be used as defined by:

$$\delta\vec{X}(t_1) = \Phi_{t_1,t_0} \delta\vec{X}(t_0) \tag{3}$$

Where $\vec{X}$ is the complete set of estimated states in the Kalman filter.

Thus the linearized measurement model becomes:

$$
\begin{aligned}
\delta\vec{\rho}(t_1) \;=\; & \left[\frac{\delta\vec{\rho}}{\delta\vec{r}_I^{\,IMU/I}(t_1)} + \frac{\delta\vec{\rho}}{\delta\vec{r}_I^{\,IMU/I}(t_0)}\Phi_{t_1,t_0}^{-1}\right]\delta\vec{r}_I^{\,IMU/I}(t_1) \;+ \\
& \delta\vec{r}_I^{\,IMU/I}(t_0) \;+ \\
& \frac{\delta\vec{\rho}}{\delta\vec{\Delta g}_{REF}^{\,REF}}\delta\vec{\Delta g}_{REF}^{\,REF} \;+ \\
& \frac{\delta\vec{\rho}}{\delta\vec{r}_I^{\,IMU/I}(t_0)}\delta\vec{r}_I^{\,IMU/I}(t_0) \;+ \\
& \frac{\delta\vec{\rho}}{\delta\vec{\theta}_{cam}}\delta\vec{\theta}_{cam}
\end{aligned}
\tag{4}
$$

Where:

$$
\frac{\delta\vec{\rho}}{\delta\vec{r}_I^{\,IMU/I}(t_1)} = -\frac{1}{|\vec{\alpha}|}\left[\vec{\alpha}^{\times}\right]^2 T_I^{cam_{t1}} \tag{5}
$$

$$
\frac{\delta\vec{\rho}}{\delta\vec{r}_I^{\,IMU/I}(t_0)} = \frac{\delta\vec{\rho}}{\delta\vec{r}_I^{\,IMU/I}(t_1)} T_{cam_{t1}}^{I} T_I^{cam_{t0}} \tag{6}
$$

$$
\frac{\delta\vec{\rho}}{\delta\vec{\Delta g}_{REF}^{\,REF}} = \left[\vec{\alpha}^{\times}\right] T_{struct}^{cam_{t0}} \tag{7}
$$

Note, $T_{struct}^{cam_{t0}}$ is the transformation matrix from the lander's structures frame to the camera's frame.

$$
\frac{\delta\vec{\rho}}{\delta\vec{\theta}_{cam}} = \left[\vec{\alpha}^{\times}\right] \tag{8}
$$

The hybrid EKF/LKF has a scalar measurement processing algorithm. Thus, measurements such as DPOS, are processed one element at a time instead of simultaneously in a vector formulation. This simplifies the navigation code base by eliminating the need for matrix inversion algorithms and provides and extra layer of robustness by completely avoiding the issue of an ill conditioned matrix inversion problem. The penalty ensued is the lack of use of the correlated measurement covariance. DPOS in particular has been shown to have error characteristics primarily normal to the direction of motion.[4] Since the measurement is formulated in the camera frame, more often than not, the DPOS noise is correlated amongst the measurement elements. However, all simulations and Monte Carlo analysis to-date have shown that ignoring these correlations in DPOS processing is sufficient to meet navigation accuracy requirements at landing.

SIMULATION TESTING

Software development for DPOS was aided by a 6 degrees of freedom (6-dof), faster than realtime simulation of the complete Nova-C flight software stack against the nominal reference trajectory. The simulation allows us to emulate all of the navigation sensors including the IMU and cameras, giving us the ability to develop and test DPOS in a software-in-the-loop system. The benefits of this include:

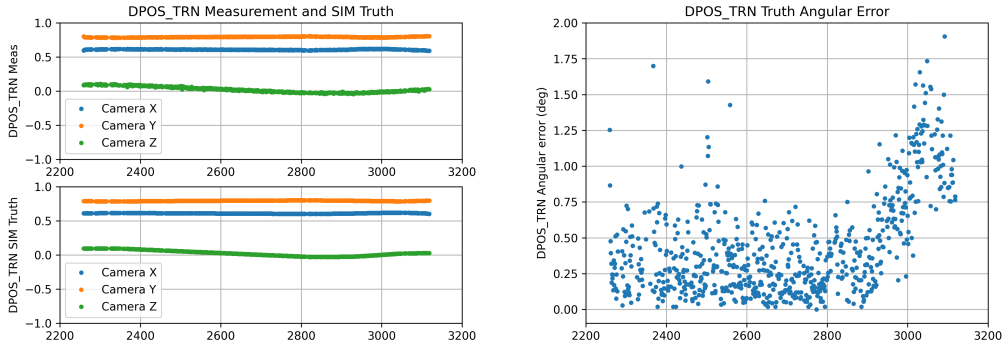
- validating the performance of the algorithm by comparing output measurements against ground truth trajectory data,

- testing the performance at various altitudes, velocities and varying terrain roughness and composition (lighting, shadows, rocks, craters, etc.),

- performing statistical analysis and algorithm parameter tuning,

- and serving as a flexible test bed for development and testing before more expensive and complex field tests

The 6-dof simulation is initialized with a known random seed, allowing us to observe the same simulated results on each subsequent run, provided that no software or configuration changes are made. To simulate the cameras, we generate a set of synthetic images based on the real camera intrinsic parameters and optical properties. The order of execution then becomes: (1) run an initial simulation with DPOS turned off and log the trajectory information, (2) generate synthetic images given the simulated trajectory, and (3) repeat the simulation run, with DPOS enabled and processing the synthetic images. This analysis process allows us to compare the processed DPOS performance vs. perfect truth data from the simulation in a realistic dynamic condition expected during flight.

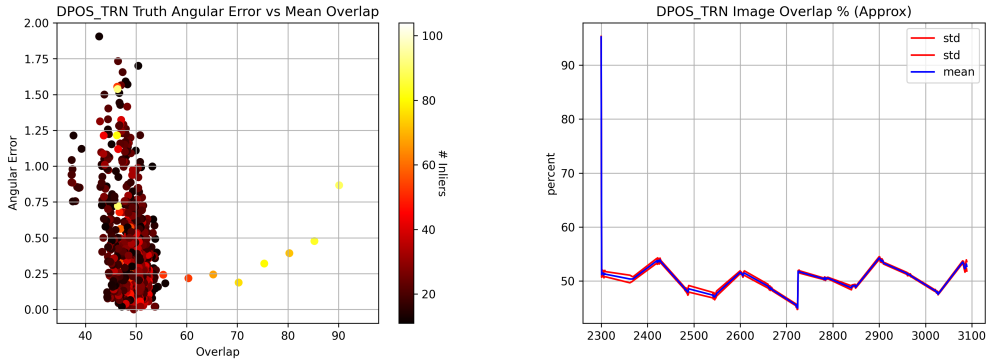


(a) Computed Measurement vs Truth Data

(b) Measurement angular error

Figure 6. DPOS measurement output and measurement error in the simulation environment.

In Figure 6(a) we plot the DPOS output vector as individual X, Y and Z direction components. In the figure you can see the direction of motion closely resembles the true direction of motion from the simulated trajectory. In Figure 6(b) we plot the cosine angle between the measured and truth DPOS measurements as a quantitative metric of error, or angular error. We would like to explain that the reason the error is increasing as the simulation time progresses is due to an inadvertent configuration change to the simulation which caused the truth 6-dof state to drift slightly from the state information used in the synthetic imagery generation process. This is a limitation of the current method of generating synthetic images, however, we are developing a solution that will eliminate the possibility of this occurring.



(a) Angular error vs Image overlap vs Number of inliers

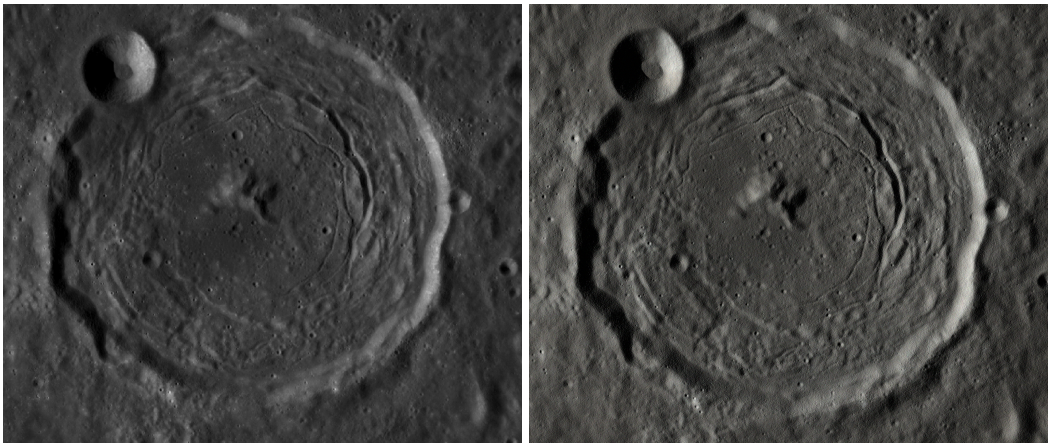
(b) Image Overlap Percentage Estimate

Figure 7. DPOS measurement error vs Image overlap and image overlap percentage in the simulation environment.

Figure 7(a) shows the angular error vs. the image overlap with the plot color indicating the number of match inliers per measurement. Through simulation testing it was determined that DPOS required a minimum number of 10 feature matches to compute a valid measurement; anything below this number of feature matches showed degraded quality of the measurement. Finally, Figure 7(b) plots the estimated image overlap percentage over the simulation run. As we mentioned in an earlier section, our baseline overlap requirement is 50% and this shows our image selection logic is working correctly and choosing images at the correct time intervals.

Synthetic Imagery

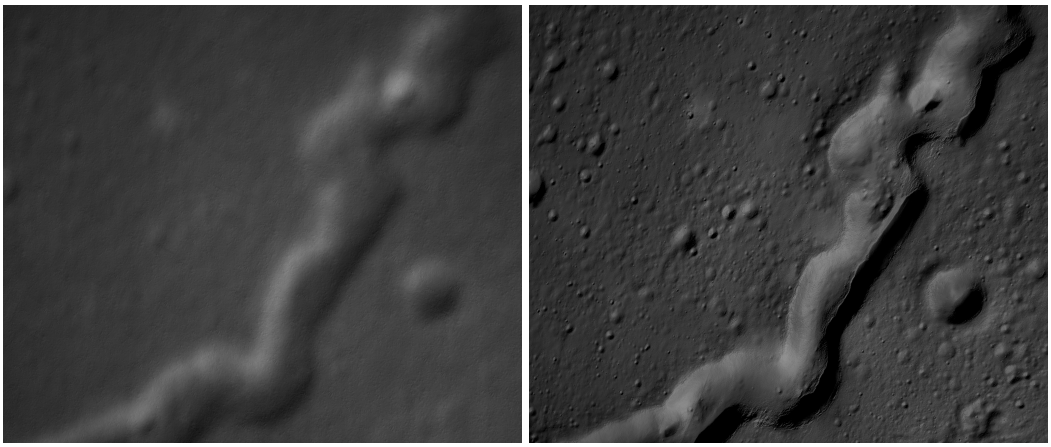
Having the ability to generate the synthetic imagery used in our simulated environment greatly influenced the way the algorithm was developed, tested, and refined. This allowed us to quickly iterate during development and stress test the performance of the algorithm. In order to have valid test data and images, our team put a great deal of effort in ensuring the quality and accuracy of the synthetic images. As part of this effort, we developed a multi-pronged process to generate synthetic images using Planet and Asteroid Natural Scene Generational Utility (PANGU)[23] and Unreal Engine 4 (UE4).[24] Additionally, we gained the ability to enhance existing digital elevation model (DEM) data with higher fidelity imagery for a final high resolution synthetic image.



(a) LROC Wide Angle Camera (b) PANGU Rendered Image

Figure 8. Images of ARISTARCHUS (23.6°N, 47.5°W) contrasting a real image and a synthetic image.

DEM data taken by LOLA[25,26] and SELENE (Kaguya)[27] are used to create accurate 3D models of lunar terrain. Color albedo maps and lunar reflectance properties are combined as part of the 3D model's material to create a realistic representation of the lunar surface. All of our synthetic imagery is generated as needed for a specific mission trajectory which has been planned, designed, and generated by the flight dynamics team. Once this is ready, the synthetic imagery team takes the trajectory data along with camera parameters based on known optics, and uses it to feed their tools and generate accurate, representative images of what our TRN and HRN cameras would observe. For high altitudes, the images are generated using PANGU as seen in Figure 8. For lower altitudes, where the DEM data has limited resolution, we enhance the DEM model with data processed and taken from Lunar Reconnaissance Orbiter (LRO) Narrow Angle Camera (NAC) imagery. We show the results of the enhancement in Figure 9.

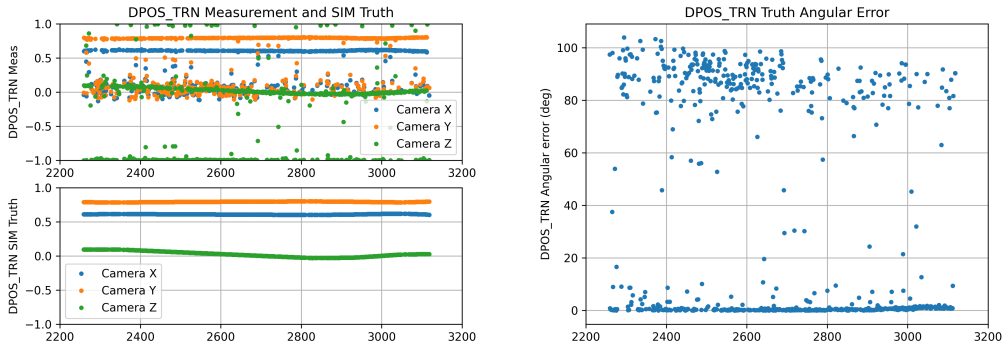


(a) Kaguya 7.4 mpp DEM (b) Kaguya 7.4 mpp DEM + NAC Image Normal Map

Figure 9. Images of TARUNTIUS (5.6°N, 46.5°E) showing the original low resolution render and the enhanced resolution render.

On the importance of outlier rejection

Thanks to the flexibility and ease of use of the simulation environment we are able to quickly change algorithm parameters and compare the resulting DPOS measurement performance. This allows us to tune and enhance the algorithm parameters for ideal performance and accuracy. In Figure 10 we show the same plots as in Figure 6, except this time we turned off RANSAC and allowed outliers to become part of the DPOS measurement solution. It becomes very clear that without outlier rejection, the DPOS measurement would not be reliable and could introduce a large amount of error into the navigation solution. In our testing, we have found that even a single outlier can have the effect of completely changing the direction of motion vector, having errors in excess of 100 degrees from the truth. We have taken great care to test and analyze our outlier rejection in order to ensure that the system will not let any outliers through.



(a) Computed Measurement vs Truth Data

(b) Measurement angular error

Figure 10. Plots of the DPOS measurement and its error in the simulation environment when robust outlier rejection is not used.

FIELD TESTING

Although the simulation environment provides a great testbed for DPOS development and algorithmic testing, it cannot expose hardware related issues such as time synchronization or camera parameter's sensitivity to natural light. Our hardware-in-the-loop NavPod system is flight-identical where possible, utilizing engineering units of our flight processors and sensors. In addition to the test hardware, we designed a variety of test procedures that would allow us to test our entire PLHA software suite. Two major field testing campaigns have been accomplished to date:

- Truck Testing
- Fixed Wing Testing

For both campaigns, engineering units of the CPU, VPU, and IMU are mounted to a structural frame. In addition, while the flight and engineering unit cameras are undergoing development and testing, a stand-in camera is utilized. When feasible by eye or range safety, engineering units of the laser range finder is also mounted to the structure. Because the engineering unit lasers are Class-4 lasers, they cannot be used in truck testing or aircraft testing in non-air restricted zones. A commercial Class-1 or "eye safe" range finder was utilized as a stand-in when required.

Colloquially, the Nova-C navigation team has referred to this test harness as the ”Navpod”. Its development has allowed for rapid deployment of navigation hardware in the loop testing, even in a terrestrial environment. We note, there is nothing inherent about the DPOS measurement that would preclude it from working in a parking lot in Houston or over the deserts of New Mexico just as well as in the lunar environment. Therefore, significant and early strides of the testing of the DPOS measurement with hardware-in-the-loop can be made with little overhead.

Truck Testing

Figure 11. Images of the truck testing setup with the Navigation Pod attached.

The first of our test campaigns is what we refer to as *truck testing*. As the name suggests, for this test we fitted a pickup truck with an aluminum frame and wench as seen in Figure 11. With the Navpod attached to the frame, the wench allowed the Navpod to move at varying velocities vertically, while driving the truck allowed for lateral motion observed by the Navpod. With independent motors controlling lateral and vertical motion a variety of trajectories could be tested, such as vertical descent, pure lateral motion at a span of altitudes, or any combination of lateral and vertical velocities.

An additional benefit of truck testing is its availability. The truck is always available to us to perform ongoing tests on at least a weekly basis, provided that the weather is acceptable for testing. We have taken advantage of this by performing a multitude of truck tests for our navigation systems and OpNav software, even being able to perform additional tests in between our other major field tests.

For all of our tests, we defined a set of coordinate frames as shown in the diagram in Figure 12. The Navpod and truck share a coordinate frame where the X-axis points up, the Y-axis points towards the right side of the truck. and the Z-axis points toward the front of the truck. The camera frame is rotated 180°on the X-axis and -45°on the Y-axis such that the Z-axis is aligned with the

boresight direction of the camera, the X-axis points up, and the Y-axis points to the right side of the camera. The image frame is the same as the camera frame, simplifying the DPOS computations.

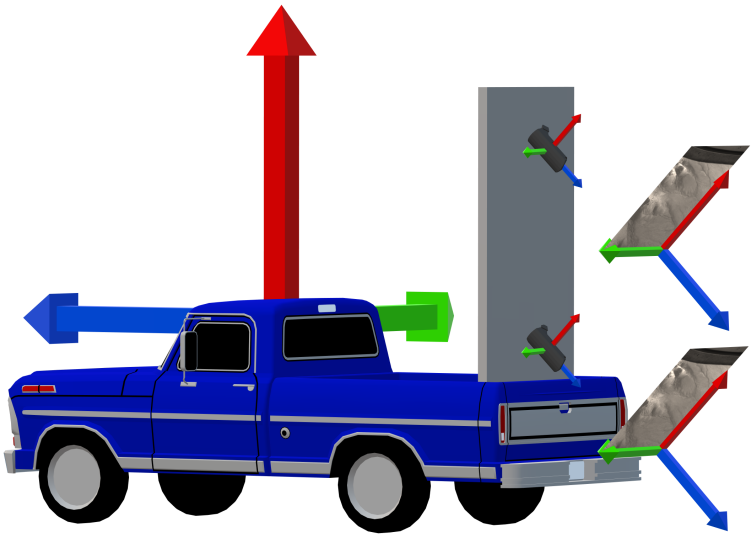


Figure 12. 3D illustration of the set of coordinate frames used in the truck test.

The main DPOS test objectives for truck testing are:

- Demonstrate DPOS real-time measurement generation with hardware-in-the-loop
- Demonstrate DPOS performance in vertical descent
- Determine optimal number of features to use in generating DPOS measurement
- Tune DPOS noise parameters and the measurement noise logic
- Tune and Test DPOS image selection logic
- Demonstrate DPOS ability to estimate lateral velocity to 0.1 m/s during vertical descent

As we transitioned from a simulation environment to a hardware-in-the-loop test rig with the Navpod and truck, we had to ensure that our software and hardware was functional and that DPOS was able to produce measurements. The very first few tests were mostly growing pains from the transition, but we were successful in running DPOS end-to-end on the Navpod. In Figure 13 we present the results of one of our vertical motion tests. In this test the Navpod starts at the bottom of the frame and is moved up towards the top followed by dropping it back to the bottom. This gives us an entirely vertical direction of motion which is clearly captured by the DPOS measurement. If we recall from Figure 12, movement towards the top as expressed in the camera frame would see most of the motion in the -Z and +X directions due to the camera moving against the camera boresight direction and up. The opposite is true when the Navpod is moving back towards the bottom of the truck. In both directions, there is no Y-axis movement since the camera remains stationary in the lateral frame.

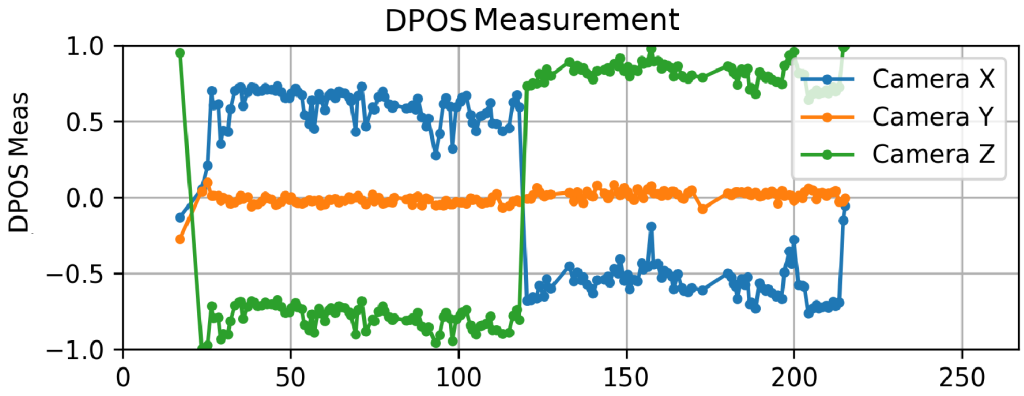
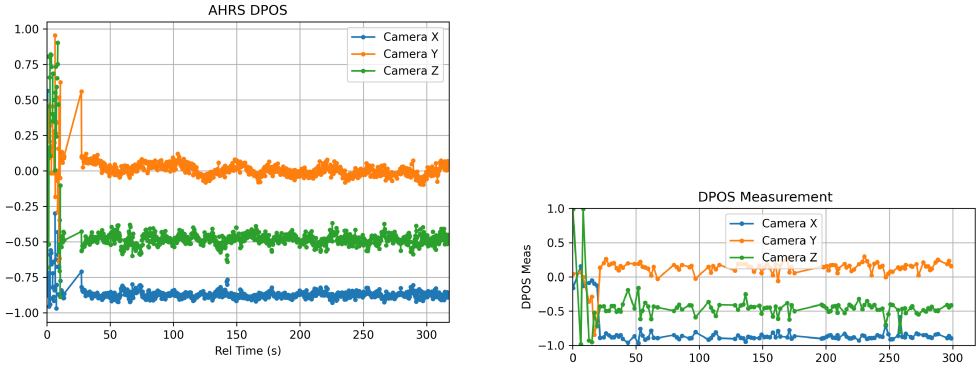


Figure 13. DPOS measurement data for a vertical motion test.

In Figure 14 we show the results of a lateral motion test in which we slowly drive the vehicle forward for approximately 5 minutes. In 14(a) we show the equivalent of the DPOS measurement as measured by an attitude and heading reference system (AHRS) device. This AHRS serves as a source of truth data to compare the computed DPOS measurement against. 14(b) shows the output of the computed DPOS measurement and we see that it matches closely to the AHRS equivalent. As you may have noticed, we did not include and AHRS comparison for the vertical motion test. This is because the AHRS is limited in its ability to measure vertical movement at slow velocities and thus it return an extremely noise signal. We have discovered additional problems with the AHRS device as a source of truth data and we are working on an alternative that will be in place for our future test campaigns.



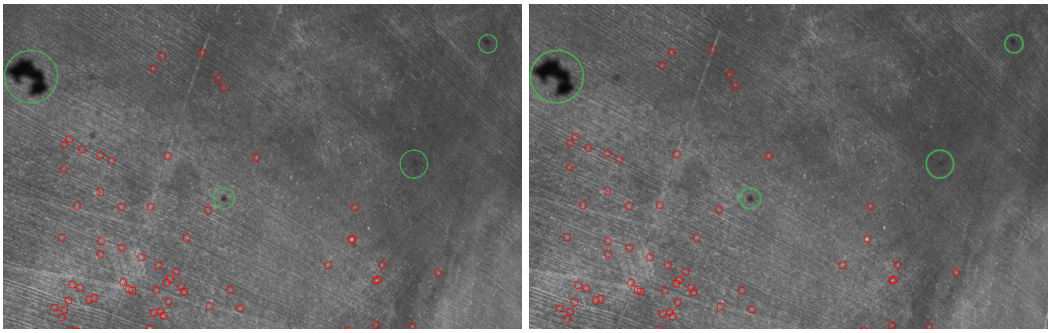
(a) AHRS Truth Data

(b) DPOS Measurement Output

Figure 14. DPOS measurement data for a lateral motion test.

An interesting test we performed in truck testing was attempting to run DPOS with a dirty camera lens. During some previous design and analysis rounds, the question was brought up on whether it would be necessary to protect the camera from debris once we are close to the surface. To mitigate for possible issues for the Nova-C lander, we decided to turn off the cameras and stop processing DPOS measurements once we reach 10m altitude. However, future missions may require us to op-

erate DPOS at lower altitudes where dust from the lunar surface could stick to the camera lens. With this in mind, in Figure 15 we present two images as they were selected for a DPOS measurement and show that none of the inlier points lies on top of the dust particulates (circled in green).



(a) Image t-1 (b) Image t

Figure 15. Images from a test that included dust on the camera lens.

Fixed Wing Testing

Figure 16. Images of the fixed wing testing setup with the Navigation Pod attached.

After truck testing, we proceeded to flying the Navpod in a fixed wing aircraft for the next phase of DPOS testing. The truck provided a test configuration to resolve preliminary hardware issues, but all testing in this configuration needed to be scaled down from that of the actual lander. For example, because the altitude above the ground was only 16ft, the lateral velocity also had to be scaled down, so 50% overlap could still be achieved. Testing in the truck also could not really test DPOS in high angular velocity scenarios. This is important to DPOS, as DPOS assumes perfect knowledge from one camera's frame to another. Testing DPOS in a higher vibration environment

was also needed as the lander is expected to experience higher vibrations than the truck provided. Aircraft testing provided us the means to test these areas.

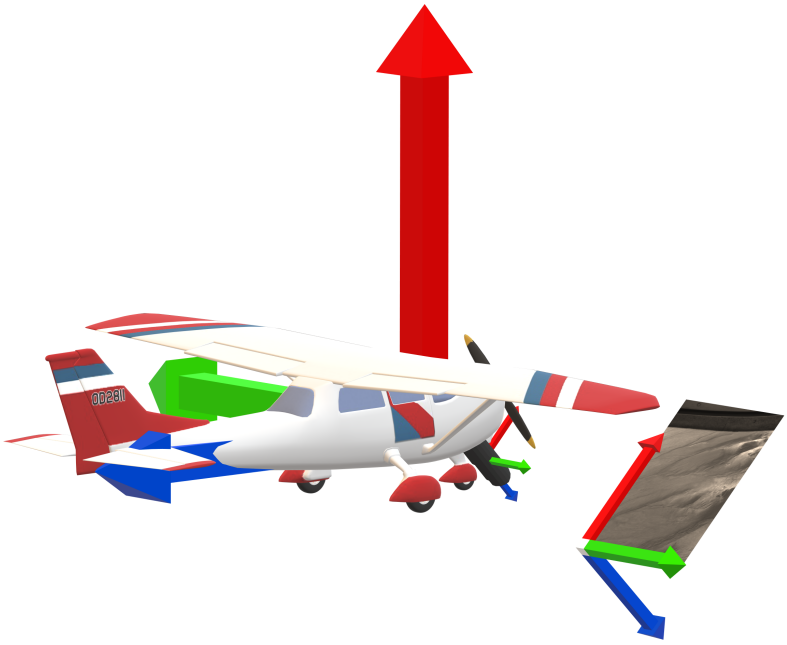


Figure 17. 3D illustration of the set of coordinate frames used in the fixed wing test.

Aircraft testing allowed the navigation system to be tested on a scale more close to the actual lander, with an increase in scaling of the test any limitations with the resolution of the camera for DPOS could be resolved. A more realistic trajectory also provided IMU and LRFS measurements and errors that are inline with what we expect to see from the actual lander.

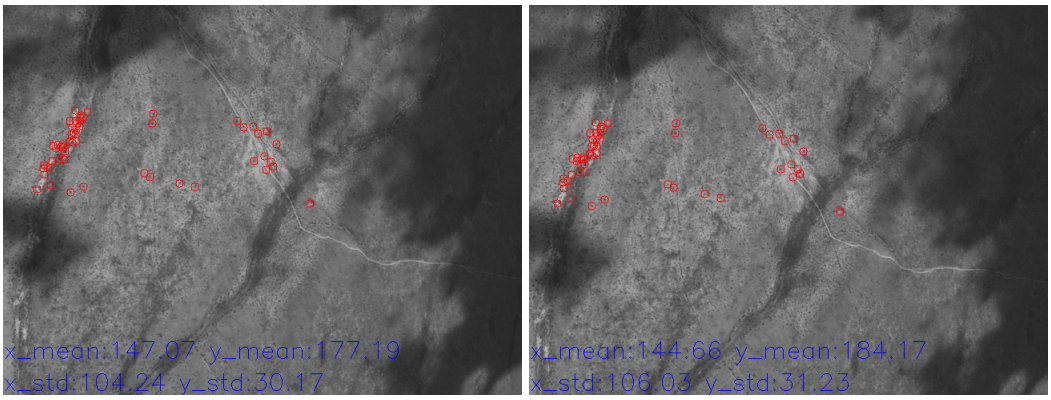
With the aircraft, DPOS could be tested at higher angular velocities. Because the truck never underwent high angular velocities, any errors in transformations from one camera frame to another were negligible. Banking back and forth in the aircraft tested the camera transformation to ensure we could assume perfect knowledge of this transformation. The aircraft also allowed us to test DPOS in a higher vibration environment. This allowed testing on the images for motion blur.

The main DPOS test objectives for fixed wing testing are:

- Prove DPOS image selection logic at high altitudes
- Prove DPOS performance in descent, horizontal, and banking profiles
- Demonstrate DPOS real-time measurement generation in a dynamic environment representative for TRN

Similar to the truck testing, we defined a set of coordinate frames for the aircraft. Shown in Figure 17, the aircraft and Navpod frames are the same and have the X-axis pointing upwards towards the ceiling of the plane, The Y-axis pointing towards the left side of the plane and the Z-axis pointing towards the tail of the plane. The camera frame is again rotated 180°on the X-axis and -45°on the

Y-axis such that the Z-axis is aligned with the boresight direction of the camera, the X-axis points up, and the Y-axis points to the right side of the camera. The image frame remains the same as the camera frame. In contrast to the truck coordinate frames, forward vehicle motion would now be expressed in the opposite direction in the camera frame.

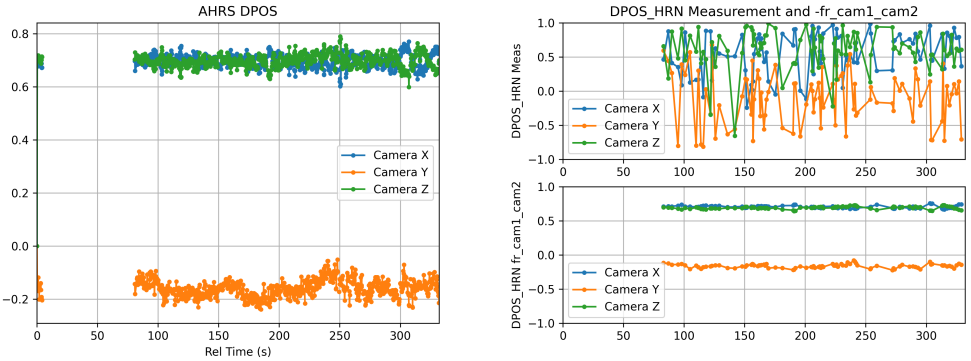


(a) Image t-1 (b) Image t

Figure 18. Images taken from the aircraft moving at high altitude.

We mentioned earlier in the paper that the DPOS measurement is a unit vector that describes the direction of motion and thus has no scale. This allows DPOS to work at any altitude and velocity as long as there are features to track in the images. This is apparent in Figure 18 where we show two sample images taken from the aircraft moving at high altitude. In both images, we see that our feature detector finds enough features to track between the two image captures to output a DPOS measurement. As we can see, it does not matter if the surface is a parking lot, a desert with shrubbery or lunar regolith, as long as our detector can find good features we will have a DPOS measurement.

Finally, we present the results of one of our test flights in Figure 19(a). Once again, we use the AHRS as a source of truth data and plot the DPOS measurement equivalent showing the direction of motion of the aircraft as expressed in the camera frame. In this instance the camera is facing towards the front of the vehicle and we see mostly equal motion in the +X and +Z axis due to the 45°off=vertical camera angle while there is a small amount of -Y axis motion. When comparing to the DPOS output in the top plot of 19(b), we notice that it is significantly noisier than the AHRS, although the general direction of motion is correct. The bottom plot in 19(b) shows the navigation system estimate of the DPOS measurement and it is more accurate. Part of the reason the DPOS measurement is noisy is due to the erratic aircraft movements. Examining multiple sets of images revealed that there is a certain side-to-side rocking of the aircraft, most likely due to high winds. However, since DPOS ingests the attitude transformation from one camera capture to the next, it should be able to account for this in the measurement.



(a) AHRS Truth Data

(b) DPOS Measurement Output

Figure 19. DPOS measurement data for a test flight.

Further investigation and testing in the truck test environment surfaced a timing synchronisation issue between the images and the rest of the navigation system. The stand-in camera system is introducing a 10-second delay in the capture and transfer of images, causing the images to have a timestamp that is approximately 10 seconds after the image was actually taken. Due to this delay, the attitude information and the image pair going into DPOS do not match and in turn cause DPOS to output a noisy measurement. Although we were able to identify the issue, it is not an issue we are able to easily fix as it appears to happen within the camera system itself. Fortunately, we have received an engineering unit for one of our flight cameras which we are in the process of integrating into the Navpod for future testing and the issue would be permanently fixed.

CONCLUSION AND FUTURE WORK

We have shared the background and software implementation details of DPOS, one of the Nova-C lander's most critical optical navigation algorithms. DPOS is a measurement which describes the lander's direction of motion between two image captures. We discussed the variety of tests performed, lessons learned and limitations from each test. In order to address some of these limitations, we are developing an advanced real-time online synthetic image generation system which will synthesize images on-demand as the simulation is running. With this system in place, every simulation will have access accurate truth states and matching images, enabling robust DPOS testing and eliminating any external sources of error related to the imagery. Finally, we look forward to our magnum opus test later this year, a helicopter test that will evaluate and stress our entire end-to-end navigation system.

ACKNOWLEDGMENT

We would like to thank our synthetic imagery team for their excellent work in the generation of realistic and accurate synthetic images. We also would like to thank our avionics and structures teams for their great effort in putting together the Navpod.

REFERENCES

[1] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004, 10.1109/CVPR.2004.1315094.

[2] D. Scaramuzza and F. Fraundorfer, "Visual Odometry Part I: The First 30 Years and Fundamentals," *IEEE Robotics & Automation Magazine*, December 2011, pp. 80–92, 10.1109/MRA.2011.943233.

[3] F. Fraundorfer and D. Scaramuzza, "Visual Odometry Part II: Matching, Robustness, Optimization, and Applications," *IEEE Robotics & Automation Magazine*, June 2012, pp. 78–90, 10.1109/MRA.2012.2182810.

[4] J. A. Christian, L. Hong, P. McKee, R. Christensen, and T. P. Crain, "Image-Based Lunar Terrain Relative Navigation Without a Map: Measurements," *Journal of Spacecraft and Rockets*, Oct. 2020, pp. 1–18, 10/gh7hgd.

[5] A. Sirotkin, "Roll Your Own Embedded Linux System with Buildroot," *Linux J.*, Vol. 2011, jun 2011.

[6] P. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces," *Procedings of the British Machine Vision Conference 2013*, British Machine Vision Association, 2013, pp. 13.1–13.11, 10/gfzbs8.

[7] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," pp. 430–443, 2006.

[8] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, Nov. 2004, pp. 91–110, 10/bqrmsp.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, Vol. 110, June 2008, pp. 346–359, 10.1016/j.cviu.2007.09.014.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision*, IEEE, Nov. 2011, pp. 2564–2571, 10/fzwvgk.

[11] O. Guclu and A. B. Can, "A Comparison of Feature Detectors and Descriptors in RGB-D SLAM Methods," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 9164, pp. 297–305, 2015. Issue: May ISSN: 16113349.

[12] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, IEEE, 2012, pp. 2681–2684, 1051-4651.

[13] A. Hietanen, J. Lankinen, J.-K. Kämäräinen, A. G. Buch, and N. Krüger, "A comparison of feature detectors and descriptors for object class matching," *Neurocomputing*, Vol. 184, Apr. 2016, pp. 3–12, 10/f8nd6n.

[14] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, Oct. 2005, pp. 1615–1630, 10/bjqbrq.

[15] E. Karami, S. Prasad, and M. Shehata, "Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images," Oct. 2017. arXiv: 1710.02726.

[16] A. B. Simon, M. Majji, C. I. Restrepo, and R. Lovelace, "A Comparison of Feature Extraction Methods for Terrain Relative Navigation," *AIAA Scitech 2020 Forum*, Reston, Virginia, American Institute of Aeronautics and Astronautics, Jan. 2020, pp. 1–12. Issue: January, 10/gh7g79.

[17] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Vol. 2018-Janua, IEEE, Mar. 2018, pp. 1–10, 10/gf3brz.

[18] Işık and K. Özkan, "A Comparative Evaluation of Well-known Feature Detectors and Descriptors," *International Journal of Applied Mathematics, Electronics and Computers*, Vol. 3, Dec. 2014, p. 1, 10/gh7g52.

[19] S. Stuart, G. Molina, and T. Crain, "THIN VPU: OPEN SOURCE VISION PROCESSING FOR SPACE NAVIGATION," AAS, 2020.

[20] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[21] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24, June 1981, pp. 381–395, 10.1145/358669.358692.

[22] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. 2004. OCLC: 171123855.

[23] I. Martin and M. Dunstan, "PLANETARY SURFACE IMAGE GENERATION FOR TESTING FUTURE SPACE MISSIONS WITH PANGU," *2nd RPI Space Imaging Workshop*, 2019, p. 13.

[24] Epic Games, "Unreal Engine," `https://www.unrealengine.com`, Dec. 2020.

[25] M. S. Robinson, S. M. Brylow, M. Tschimmel, D. Humm, S. J. Lawrence, P. C. Thomas, B. W. Denevi, E. Bowman-Cisneros, J. Zerr, M. A. Ravine, M. A. Caplinger, F. T. Ghaemi, J. A. Schaffner, M. C. Malin, P. Mahanti, A. Bartels, J. Anderson, T. N. Tran, E. M. Eliason, A. S. McEwen, E. Turtle, B. L. Jolliff, and H. Hiesinger, "Lunar Reconnaissance Orbiter Camera (LROC) Instrument Overview," *Space Science Reviews*, Vol. 150, Jan. 2010, pp. 81–124, 10.1007/s11214-010-9634-2.

[26] M. Barker, E. Mazarico, G. Neumann, M. Zuber, J. Haruyama, and D. Smith, "A new lunar digital elevation model from the Lunar Orbiter Laser Altimeter and SELENE Terrain Camera," *Icarus*, Vol. 273, July 2016, pp. 346–355. Publisher: Elsevier Inc., 10/f8n6tp.

[27] H. Araki, S. Tazawa, H. Noda, Y. Ishihara, S. Goossens, S. Sasaki, N. Kawano, I. Kamiya, H. Otake, J. Oberst, and C. Shum, "Lunar Global Shape and Polar Topography Derived from Kaguya-LALT Laser Altimetry," *Science*, Feb. 2009. Publisher: American Association for the Advancement of Science, 10.1126/science.1164146.