

## EVALUATING IMAGE PROCESSING ALGORITHMS FOR DRAGONFLY OPTICAL NAVIGATION

**Steve N. Jenkins\*, Ike R. Witte\*, Kevin H. Foster\*, Gunner S. Fritsch\*, Samuel E. Bibelhauser\*, Ben M. Schilling\*, Timothy G. McGee†**

NASA's *Dragonfly* mission involves an autonomous rotorcraft traversing the skies over Titan. The Electro-optical Terrain Sensing (ETS) function of the rotorcraft's mobility subsystem generates optical measurements using a downward facing camera for the onboard navigation algorithms. These terrain-relative measurements, in part, provide the capability for safe flight, allowing the rotorcraft to relocate itself to new scientifically interesting sites on Titan.

Recent modifications to the ETS design, discussed in this paper, enhance the flexibility and robustness of the algorithm. Three variations—full image correlation, batch-processed image patch correlations, and filtering-independent image patch correlations—are compared in terms of their overall impact on the navigation filter. Considerations encompass sensitivity to knowledge error, robustness to environmental factors, resource utilization, and overall measurement error distribution. The findings contribute to the advancement of *Dragonfly*'s mobility subsystem for navigating Titan's unique terrain with greater efficiency and adaptability.

### INTRODUCTION

NASA's *Dragonfly* mission is designed to autonomously fly a rotorcraft across the surface of Titan, make scientific measurements, and sample the moon at scientifically interesting sites identified by operators on Earth.<sup>1,2</sup> The Electro-optical Terrain Sensing (ETS) function of *Dragonfly*'s Mobility subsystem generates optical measurements for the onboard navigation function, enabling in-flight terrain-relative navigation without a priori reference maps. This capability enables the Lander to autonomously traverse unknown terrain, track previously flown trajectories, and precisely land in areas designated hazard-free by the Mobility subsystem.

The ETS design builds on the Autonomous Precision Landing Navigation Algorithm (APLNav) developed during the Autonomous Landing and Hazard Avoidance Technology (ALHAT) program.<sup>3</sup> *Dragonfly* faces two additional challenges to those tackled during APLNav's development. First, the majority of flights are expected to take place over low feature terrain similar to terrestrial sand dunes. Second, the terrain height variation over an image field of view will at times be relatively large compared to the altitude. Significant modifications have been made to operate in these conditions and without detailed maps of the terrain, as well as to efficiently make use of *Dragonfly*'s available computational resources, specifically the Onboard Vision Processor (OVP) FPGA. Detailed analysis of the performance of the design using sensor and environment models, as well as testing with earth analogues, has been performed to ensure the adapted algorithm achieves the system requirements for reliable surface navigation.

The result is a design robust to the uncertainty present in our understanding of the Titan environment, and able to accommodate modifications through parameter uploads to tune the optical

---

\*Johns Hopkins Applied Physics Laboratory, 11100 Johns Hopkins Road Laurel, MD 20723

†Point Mass Technologies, Pittsburgh, PA

processing procedure as more is learned after entry, descent, landing, and through successive flights on the surface. While Titan has been explored from afar by *Voyager*, *Cassini*, and Earth-based telescopes, and briefly in situ with the *Huygens* probe, the surface has only been mapped at scales much larger than the length scales relevant to terrain-relative navigation image processing algorithms. *Dragonfly* will be a true explorer, traversing landscapes never before seen by people or machines.

## ALGORITHM DESCRIPTION

The ETS algorithms produce velocimetry and localization measurements for the onboard navigation filter using phase-only cross-correlation to align portions of two images captured with a navigation camera (NavCam) built by Malin Space Science Systems. A velocimetry measurement is generated by processing two contemporaneous images, while a localization measurement compares a newly captured image to one captured earlier in the flight or on a previous flight. These reference images from the past are referred to as breadcrumbs, and are critical in reducing filter drift and navigating relative to a selected landing site.

A prior ETS design, targeted to *Dragonfly*'s computing architecture at the time, involved aligning the entirety of the two images in a single correlation, and is described in a recent paper.<sup>4</sup> A more capable FPGA, the Kintex KU060, has since been selected for the OVP, enabling additional image processing resources. The following sections detail the current design with the modifications made in order to take advantage of these resources. The implementation can be split into four main components executed in series: OVP Preprocessing, ETS Selection, OVP Correlation, and ETS Measurement Generation.

### OVP Preprocessing

OVP Preprocessing preprocesses a newly acquired NavCam image and finds potential areas of interest to match to previous imagery. The raw image read off the detector is 2048 x 2048 pixels, with a field of view of near 90 deg. The image is 2x2 binned on the camera to 1024 x 1024 and read out onto the OVP FPGA, where the fixed pattern noise (FPN) is removed through a subtractive dark frame and multiplicative flat-field correction, as shown in Eqs. (1)–(2).  $\mathbf{I}$  is the raw image,  $\mathbf{D}$  is the dark frame,  $\mathbf{F}$  is the flat-field,  $\langle \mathbf{F} - \mathbf{D} \rangle$  is the mean of the flat-field minus the dark frame, and  $\mathbf{I}'$  is the corrected image.

$$\mathbf{I}'(x, y) = \left( \mathbf{I}(x, y) - \mathbf{D}(x, y) \right) \times \mathbf{G}(x, y) \quad (1)$$

$$\mathbf{G}(x, y) = \frac{\langle \mathbf{F} - \mathbf{D} \rangle}{\mathbf{F}(x, y) - \mathbf{D}(x, y)} \quad (2)$$

The resultant calibrated image is then 2x2 binned again to create a 512x512 image, and areas of interest are determined by computing the Harris corner response. The Harris corner response,  $\mathbf{C}$ , shown in Eqs. (3)–(4) is a heuristic used to identify image patches that should correlate well by quantifying the sharpness of their auto-correlation peak.<sup>5</sup>  $\mathbf{M}$  is the structure tensor of the image,  $k$  is an empirical parameter set to 0.05 by default, and  $\mathbf{I}_x$  and  $\mathbf{I}_y$  are the derivatives of the image in the x and y direction. The summation is weighted over a window around each pixel.

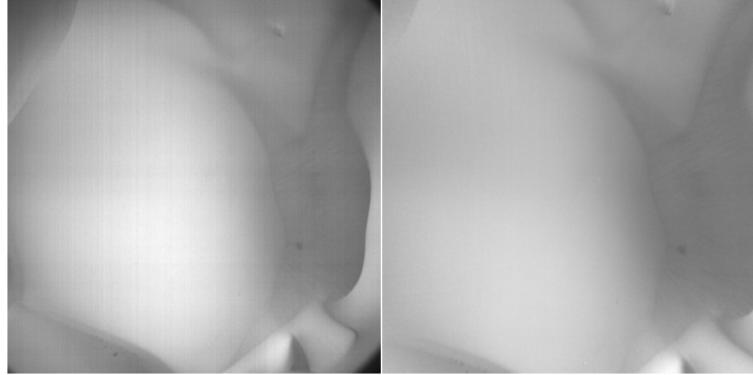
$$C = \det(\mathbf{M}) - k * \text{trace}(\mathbf{M})^2 \quad (3)$$

$$\mathbf{M} = \sum_{(x,y) \in W} \begin{bmatrix} \mathbf{I}_x^2 & \mathbf{I}_x \mathbf{I}_y \\ \mathbf{I}_x \mathbf{I}_y & \mathbf{I}_y^2 \end{bmatrix} \quad (4)$$

Spatially distributed patch correlations are enforced by applying non-maximum suppression (NMS) to select local maxima in the Harris response matrix, in order to add geometric diversity and avoid wasted computation on near-identical patches. NMS is implemented distributed across the OVP FPGA and ETS software. OVP Preprocessing first reduces the information by determining the maximal Harris response location in each 8x8 grid square of the 512x512 matrix. Then ETS selects patches from these 64 locations using a priori knowledge of the expected overlap in the images.

After the Harris corner response is computed, OVP normalizes the image by subtracting the image mean, and then removes the significant distortion inherent in the wide field of view NavCam lens. Inverse warping using a 6th order Brown's distortion model, detailed in Eq. (5), is applied to avoid gaps in the corrected image.<sup>6</sup> The effective field of view of the flattened image is expected to be near 80 deg.  $x_{\text{corrected}}$  and  $y_{\text{corrected}}$  are the rectified pixel locations of the raw pixel locations  $x$  and  $y$ ,  $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ ,  $(x_0, y_0)$  is the principal point of the camera, and  $k_1$ ,  $k_2$ , and  $k_3$  are the calibrated distortion coefficients. An example of a raw NavCam image and its corrected counterpart from a flight test over the Imperial Sand Dunes is shown in Figure 1.

$$\begin{aligned} x_{\text{corrected}} &= (x - x_0)(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + x_0 \\ y_{\text{corrected}} &= (y - y_0)(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + y_0 \end{aligned} \quad (5)$$



**Figure 1** Comparison of a raw image (left) during a flight test over desert terrain and the image after fixed pattern noise removal and distortion correction (right).

### ETS Selection

While modern terrestrial autonomous robotics employ the use of simultaneous localization and mapping (SLAM),<sup>7</sup> due to the computational constraints of *Dragonfly*'s RAD750 flight computer,

the navigation filter is unable to carry all of the states necessary. Instead, ETS and the navigation filter use a "breadcrumb" method that buffers images and associated states distributed over the trajectory. These breadcrumbs can be thought of as similar to SLAM keyframes without their associated cross-covariances with the current navigation state, and the overall method can be considered a "SLAM-Lite" variant. Reference images are selected by approximating the scale offset and overlap of each buffered image with the current image. A recently captured image with the largest amount of overlap is chosen for the velocimetry reference image, and the oldest breadcrumb estimated to be close enough for correlation is chosen as the breadcrumb reference image. ETS Selection selects the images to correlate, computes the homographies that relate the two a priori estimated image frames, and determines the patch locations to correlate.

Because image correlation is inherently a two dimensional operation, a homography is used to remove known angular and scale differences by warping the current image into the reference image frame using Eq. (6) prior to correlation.

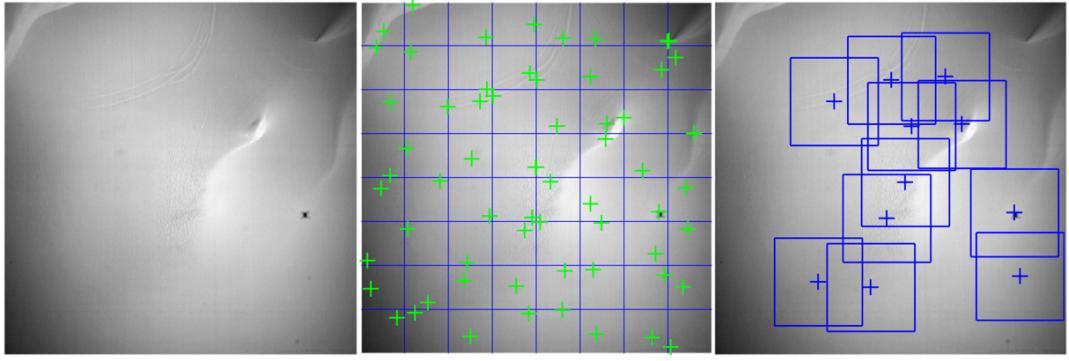
$$\mathbf{H}_r^c = \mathbf{A} \mathbf{R}_{NED}^c \left( \mathbf{I}_{3x3} - \frac{\mathbf{t}_r^c}{d_c} \mathbf{n}^T \right) (\mathbf{R}_r^{NED}) \mathbf{A}^{-1} \quad (6)$$

$\mathbf{R}_{NED}^{c,r}$  is the rotation matrix from the North-East-Down (NED) frame to the current ( $c$ ) or reference ( $r$ ) camera frame,  $\mathbf{I}_{3x3}$  is the identity matrix,  $\mathbf{t}_r^c$  is the translation vector between reference and current camera frames,  $d_c$  is the perpendicular ground distance from the terrain to the current camera frame,  $\mathbf{n}$  is the best estimate of the normal vector of a plane fit to the terrain inside the field of view, and  $\mathbf{A}$  is the camera intrinsic matrix. The terrain normal vector and perpendicular ground distance are estimated in the navigation filter and driven by LiDAR measurements.  $\mathbf{A}$  is composed of the principal point  $(x_0, y_0)$  and the principal distance  $p_d$ , shown in Eq. (7).

$$\mathbf{A} = \begin{bmatrix} p_{dx} & 0 & x_0 \\ 0 & p_{dy} & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

After the reference images for velocimetry and breadcrumb measurements are selected and the homographies are calculated, the 64 Harris corners are downsampled to a desired number of patches for each measurement type. The Harris corners are first sorted by their response strength. Then the strongest corner inside a border around the image, and within a region defined by the expected overlap in the two images, is selected. This is to ensure at most only a small portion of the image patch around the Harris corner will be devoid of data after being warped into the opposing image frame. Any of the 63 remaining Harris corners within a parameterized region nearby this selected corner are removed from consideration, and the process is repeated until the chosen number of patches are determined. This process completes the NMS began in OVP Preprocessing, and is repeated both for the velocimetry and breadcrumb measurements. Figure (2) details the OVP/ETS process to select several patches to correlate.

In addition to selecting the set of patch locations to correlate in the velocimetry and breadcrumb measurements, ETS also designates one of several pre-loaded windows to apply to each image patch, and another to apply to the reference image. These windows are stored on the OVP as vectors of length 512, and computed as the outer product of these vectors as needed. The three windows currently in use are a 512x512 Hann window for full image correlation, and both a 194x194 and 512x512 Tukey window for the patch and reference image respectively in patch correlations. The



**Figure 2** Harris corner example during an ITP flight over the Imperial Dunes, CA, showing the raw image (left), the 8x8 grid of Harris peaks OVP reports (center), and the subset of patches selected by ETS for correlation (right).

smaller Tukey window is shifted to the center of the patch undergoing correlation, and zero-padded to 512x512 to simplify the OVP implementation. Windowing the imagery in this way avoids image correlation artifacts from edge effects.

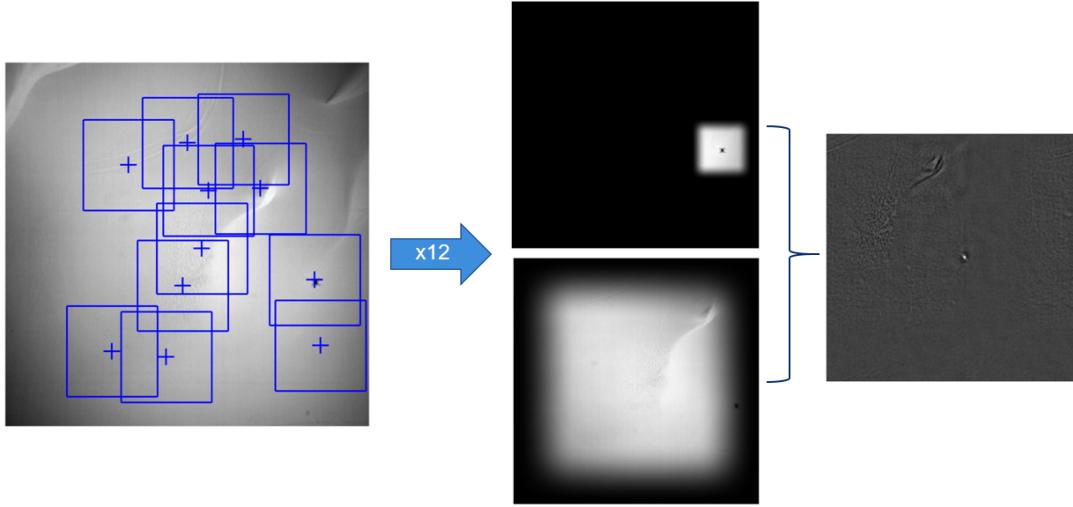
### OVP Correlation

The OVP Correlation design has largely remained the same with the addition of the capability to perform multiple correlations within a second. The total number of correlations achievable is in flux as the system design matures. OVP retrieves the reference images for the velocimetry and breadcrumb measurements, and warps the preprocessed current camera image into the reference image frames using inverse homographic warping. As in the case of distortion correction, performing inverse warping avoids gaps in the warped image and is accomplished using bilinear interpolation. Following warping, the desired windowing coefficients are shifted to align with the chosen patch location, multiplied to generate a matrix, and the matrix is applied to the image to set up a correlation. The reference image has a separate window applied that reduces the signal towards the edges, but leaves the majority of the image untouched. Once both the current and reference images are windowed, a phase-only cross correlation is performed to match the signal from the warped current image patch to the signal of the reference image, using Eq. (8).

$$C = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{I_1\} \otimes \mathcal{F}\{I_2\}}{\|\mathcal{F}\{I_1\} \otimes \mathcal{F}\{I_2\}\|} \right\} \quad (8)$$

$\mathcal{F}$  denotes the Fourier transform,  $I_1$  and  $I_2$  are the two images to be correlated, and  $\otimes$  denotes element-wise multiplication. A low-pass filter is applied while in the frequency domain to the correlation surface to avoid very high frequency content. The team is currently investigating band-pass filters to replace the blanket low-pass filter. Figure (3) depicts the patch selection, windowing, and resultant correlation matrix from an image correlation.

The location of the primary peak of the correlation surface indicates the shift that most strongly matches the signal of the two images. Once the peak location is found, a neighborhood around it is set to be zero, and the process is repeated to find the secondary peak value and location. These data



**Figure 3** Example set of patch correlations between a subset of the current image and the full reference image modified with Tukey windows and showing the correlation surface.

are used to assess the image correlation's quality, and generate a measurement for the navigation filter.

### ETS Measurement Generation

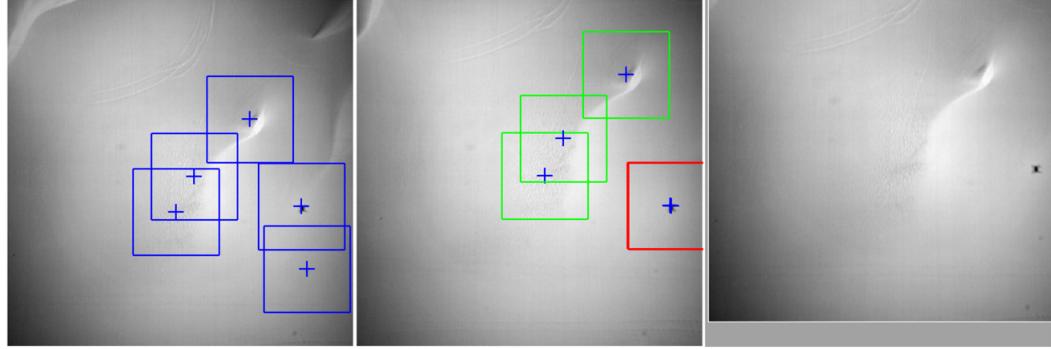
ETS Measurement Generation's primary function is to translate the correlation results from OVP into measurements to be ingested by the navigation filter. The process requires selecting the correlations that show a decisive signal, removing outliers, and packaging up the results.

ETS uses the secondary-peak-to-peak ratio, shown in Eq. (9), as a measure of image correlation strength. All correlations that have a ratio above a set threshold are removed from further processing due to the ambiguity in the shift to best align the patch with the reference image. The peak locations of the valid correlations, on the other hand, are refined through a parabolic fit to the 3x3 neighborhood around the peak to estimate a subpixel maximum.

$$peak_{ratio} = \frac{peak_{secondary}}{peak_{primary}} \quad (9)$$

The resultant correlations are then checked for geometric consistency, assuming the ground is a flat plane, and any geometric outliers are rejected. The method to do this is similar to modern feature tracking techniques using random sample consensus but is done exhaustively here due to the small number of patches. The number of patches needed to compute the desired transform is randomly selected (one for 2DOF, two for 3DOF) and the transform is calculated. Using the computed transform, the other patch locations are reprojected and the residual error between the projected patch location and where it was actually matched is computed. Those patches projected to within a threshold of the matched locations are marked as inliers, and if the number of inliers is larger than the current maximum, the estimated transform is updated. By iterating through all possible combinations, the final estimated transform will result in the largest number of inliers, and a least-squares estimation is performed using all inliers. This process is performed for both the

2DOF and 3DOF cases. Figure (4) shows an example of an optical artifact, a drone self-shadow not expected on Titan due to the diffuse illumination, correctly rejected as an outlier.



**Figure 4** Visualization of the geometric outlier rejection that is available with multiple patch correlations.  $Img_1$  is shown with the patches that passed the peak ratio check (left),  $Img_2$  is shown with the corresponding matched patches labeled as inliers (green) and outliers (red), followed by  $Img_1$  warped into the frame of  $Img_2$  based on the estimated transform from the patch correlation results.

Examples of 2DOF and 3DOF transforms, as well as a 4DOF similarity transform that includes rotation, are shown in Eq. (10).  $t_x$  and  $t_y$  are the translations in column and row, respectively,  $s$  is the scale change, and  $\theta$  is the image rotation.

$$\mathbf{T}_{2dof} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}_{3dof} = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}_{4dof} = \begin{bmatrix} s \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & s \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

## Measurements and Measurement Models

An earlier version of ETS's measurement model was formulated as a delta position measurement based upon a single full-image correlation. This has been updated to a more tightly-coupled approach between ETS and the navigation filter using multiple image patch correlations. ETS outputs matching pixel locations between the two images to the navigation filter instead, akin to SLAM approaches, which allows the filter to better account for state and measurement uncertainties.

The measurement model for the tightly-coupled solution begins with the equations of projective geometry, which detail the projection of a 3D ray onto the 2D image plane. This is shown in Eq. (11), where  $\mathbf{x}_p^{NED}$  is the location of point p in the North-East-Down (NED) frame,  $\mathbf{R}_{NED}^{c,r}$  is the rotation matrix from the NED frame to the camera frame,  $(i_p, j_p)$  is the pixel location on the image plane for general point p, and  $z_p$  is the distance from the camera to point p, and  $S_{1,2}$  is a selector matrix for the first two elements in a vector.

$$\begin{bmatrix} i_p \\ j_p \end{bmatrix} = \frac{1}{z_p^{Cam}} S_{1,2} \mathbf{A} \mathbf{x}_p^{Cam} = \begin{bmatrix} pdx \frac{x_p^{Cam}}{z_p^{Cam}} + x_0 \\ pdy \frac{y_p^{Cam}}{z_p^{Cam}} + y_0 \end{bmatrix} = \frac{1}{z_p^{Cam}} S_{1,2} \mathbf{A} \mathbf{R}_{NED}^{Cam} (\mathbf{x}_p^{NED} - \mathbf{x}_p^{Cam}) \quad (11)$$

Eq. (12) reiterates the projective geometry equation, with  $(t_r)$  specifying that point p (the location on the ground that is the center of the patch) projects to the reference frame pixel location  $(i_r, j_r)$  for the pose at time  $t_r$ . ETS is warping the current frame into the reference, so  $(i_r, j_r)$  is equivalent to the point pixel location in the current frame  $(i_c, j_c)$  warped into the reference frame using the homography  $H_c^r$ .

$$\begin{aligned} \begin{bmatrix} i_r \\ j_r \end{bmatrix} &= \frac{\mathbf{S}_{1,2}\mathbf{A}\mathbf{R}_{NED}^{Cam}(t_r)}{z_p(t_r)} \left( \mathbf{x}_p^{NED} - \mathbf{x}_{CAM}^{NED}(t_r) \right) \\ &= \mathbf{H}_c^r \begin{bmatrix} i_c \\ j_c \end{bmatrix} = \mathbf{H}_c^r \frac{\mathbf{S}_{1,2}\mathbf{A}\mathbf{R}_{NED}^{Cam}(t_c)}{z_p(t_c)} \left( \mathbf{x}_p^{NED} - \mathbf{x}_{CAM}^{NED}(t_c) \right) \end{aligned} \quad (12)$$

This is further expanded to factor in the change in position between times  $t_c$  and  $t_r$ , projected into the reference image frame along with the patch center in the current image frame warped into the reference image frame.  $\eta_{corr}$  represents the white noise from the correlation that matches the patches, while  $\eta_z$  represents additional noise used to compensate for extrapolating the perpendicular ground distance (PGD) outside the LiDAR FOV to model  $z_p(t_c)$ .

$$\begin{aligned} \begin{bmatrix} i_r \\ j_r \end{bmatrix} &= \frac{\mathbf{S}_{1,2}\mathbf{A}\mathbf{R}_{NED}^{CAM}(t_r)}{z_p(t_r)} \left( \mathbf{x}_{CAM}^{NED}(t_c) - \mathbf{x}_{CAM}^{NED}(t_r) \right) \\ &\quad + \frac{z_p(t_c)}{z_p(t_r)} \mathbf{S}_{1,2}\mathbf{A}\mathbf{R}_{NED}^{CAM}(t_r) \mathbf{R}_{CAM}^{NED}(t_c) \mathbf{A}^{-1} \begin{bmatrix} i_c \\ j_c \\ 1 \end{bmatrix} + \eta_{corr} + \eta_z \end{aligned} \quad (13)$$

Because the 15 degree LiDAR FOV is smaller than the image, the range to each pixel is not explicitly measured for each image and the range therefore must be based on the slope estimates ( $\hat{\mathbf{n}}^{NED}$ ) from the navigation filter as well as the PGD,  $(d)$ , which is the shortest distance from the terrain to the camera.

$$z_p(t_r) = [0 \ 0 \ 1] \cdot \mathbf{R}_{NED}^{Cam}(t_r) \left( z_p(t_c) \mathbf{R}_{CAM}^{NED}(t_c) \mathbf{A}^{-1} \begin{bmatrix} i_c \\ j_c \\ 1 \end{bmatrix} + \mathbf{x}_{CAM}^{NED}(t_c) - \mathbf{x}_{CAM}^{NED}(t_r) \right), \quad (14a)$$

where

$$z_p(t_c) = \hat{\mathbf{l}}_p^{Cam}(t_c)(3) \frac{d(t_c)}{\hat{\mathbf{l}}_p^{Cam}(t_c) \cdot \hat{\mathbf{n}}^{NED}(t_c)} \quad (14b)$$

$$\hat{\mathbf{l}}_p^{Cam} = \mathbf{A}^{-1} \frac{[i_p \ j_p \ 1]^T}{\sqrt{1 + i_p^2 + j_p^2}}. \quad (14c)$$

## Implementation

*Dragonfly*'s computing architecture necessitates much of the resource-intensive image processing to occur on the OVP FPGA in order to achieve a 1 Hz measurement cadence. Due to the selection of

the Kintex KU060 FPGA, the current system can now support generating ETS measurements with multiple image correlations per second. Through simple parameter changes to define the image window applied before correlation, the updated design presented here can implement the original full-image correlation algorithm, or the new method that correlates smaller sections of the incoming NavCam image to the reference image. Smaller patches tend to provide more robustness to scale and heading knowledge errors, while larger patches retain more terrain signal to match lower spatial frequency features, and thus tend to be able to perform better in bland terrain. Material presented in later sections details the analysis to explore the optimal patch size given the present knowledge of the Titan environment.

## MODELING

High quality models have been critical in order to gain confidence in the algorithmic design, and explore the trade space between full-image and smaller patch correlations. A combination of environmental physics-based models and models generated using data from Earth analogues, as well as imagery collected from flights with an terrestrial test platform, have been essential in developing and analyzing the algorithms.<sup>8,9</sup> These models capture the uncertainty present in our understanding of Titan, and the ETS function has been designed to be robust to these variations. Moreover, the ability to adapt to the environment on Titan, after entry, descent, and landing, has been a strong consideration in the algorithmic implementation. The models necessary to generate synthetic imagery have been collected into JHUAPL's Renderer and Camera Emulator (RCE) ray-tracing software.

The main underlying digital terrain model (DTM) used for ETS performance analysis was constructed using stereo satellite imagery of the Namib desert, and post-processed to remove vegetation and other artifacts. The Namib was selected as a close analogue to the Shangri-La Dunes: *Dragonfly*'s initial destination at Titan.<sup>10</sup> Various levels of perlin noise and rock size-frequency models are applied to the Namib DTM to provide terrain resolution below the level at which the stereo processing could achieve, to explore image correlation sensitivities, and to ensure algorithmic robustness to a wide range of conditions. As a conservative measure, rocks are commonly removed from the model entirely during ETS analyses.

A lighting model was generated using data on Titan's atmosphere from the *Cassini-Huygens* mission. Titan's distance from the sun, and the haze in the moon's atmosphere make these models critical to understanding the expected terrain signal in the imagery, as well as the necessary camera exposure time. The details of generating this lighting model is described in a previous paper.<sup>4</sup> The general effect of the lighting environment on Titan is a dim, diffusely lit landscape, such as at dusk on Earth.

Finally, a camera model was generated with the specs of Malin Space Science Systems' NavCam, augmented with data collected through internal testing of a NavCam development model. The model reproduces the optical and electronic effects of the NavCam such as lens distortion, shot noise, photo response non-uniformity, and analog-to-digital conversion.

## Integrated Test Platform

In addition to software models of *Dragonfly* and Titan, the Dragonfly team built an Integrated Test Platform (ITP) octocopter designed for terrestrial flight with which to perform various Mobility system integration tests, and guidance, navigation, and control analyses. The ETS team used this

platform to acquire imagery with scaled down flight profiles in California's Imperial Sand Dunes.<sup>9</sup> These data contain images that are quite bland, and provide an independent method to analyze the robustness of ETS's design and implementation.

## ANALYSIS

This section compares three algorithmic variations that are achieved through parameter changes to the flight software: full image correlation, batch-processed image patch correlations, and filtering independent image patch correlations. Considerations include measurement error statistics, sensitivity to a priori knowledge error, and robustness to environmental factors. The impact of each approach on the navigation filter is discussed.

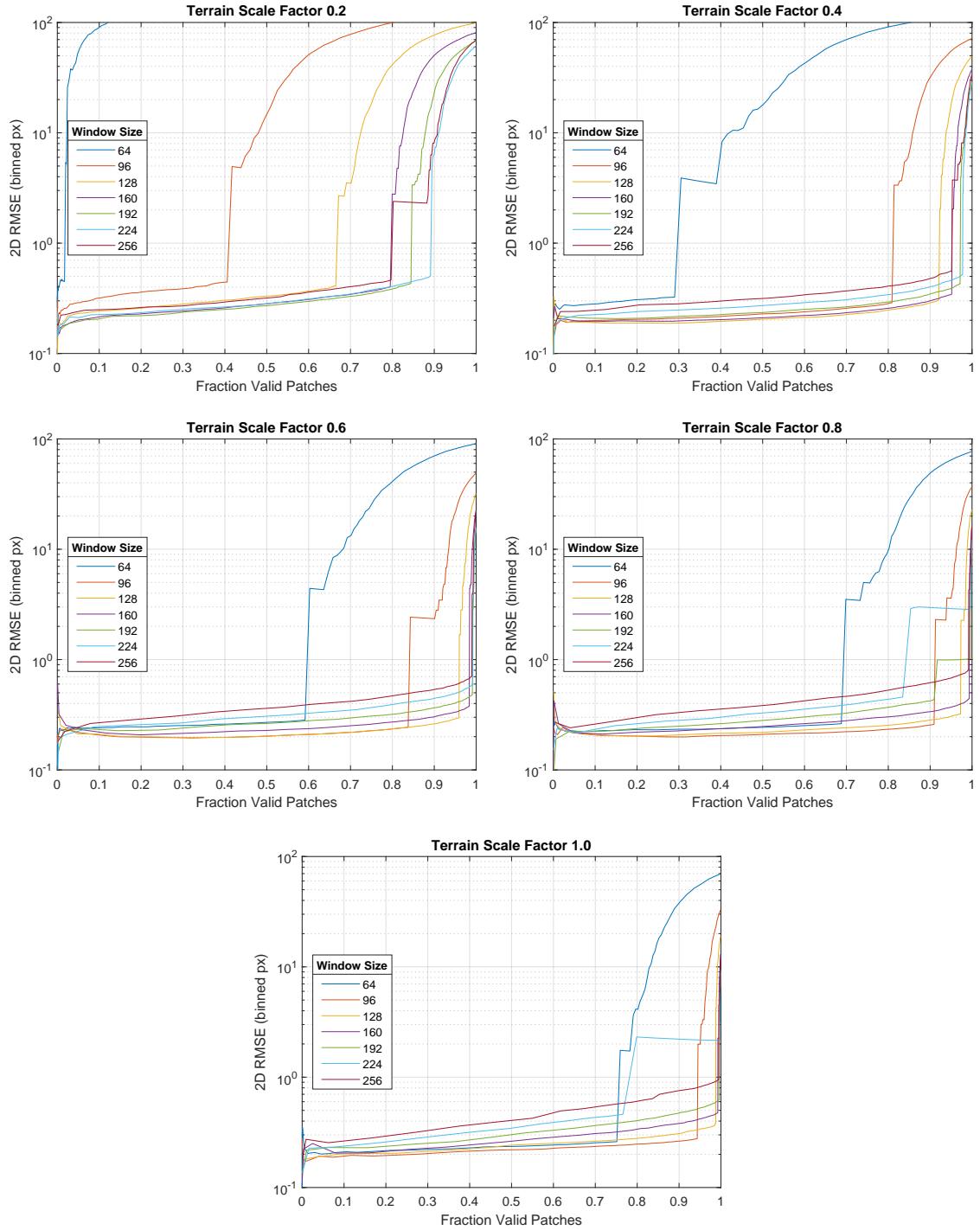
### Robustness to Environmental Factors

Environmental factors will contribute to the difficulty in successfully correlating two images and extracting accurate terrain-relative measurements. The main sources of environmental unknowns are the possibility of bland terrain and the relief/slope present in the terrain morphology. ETS relies upon having enough signal present in the imagery to provide a strong correlation signal between two images. Terrain that is significantly blander than expected may result in regions where there are no strong peaks in the correlation; unaccounted for relief/slope can cause the correlation peaks to spread out and divide, as each feature in the image exhibits a different translation offset.

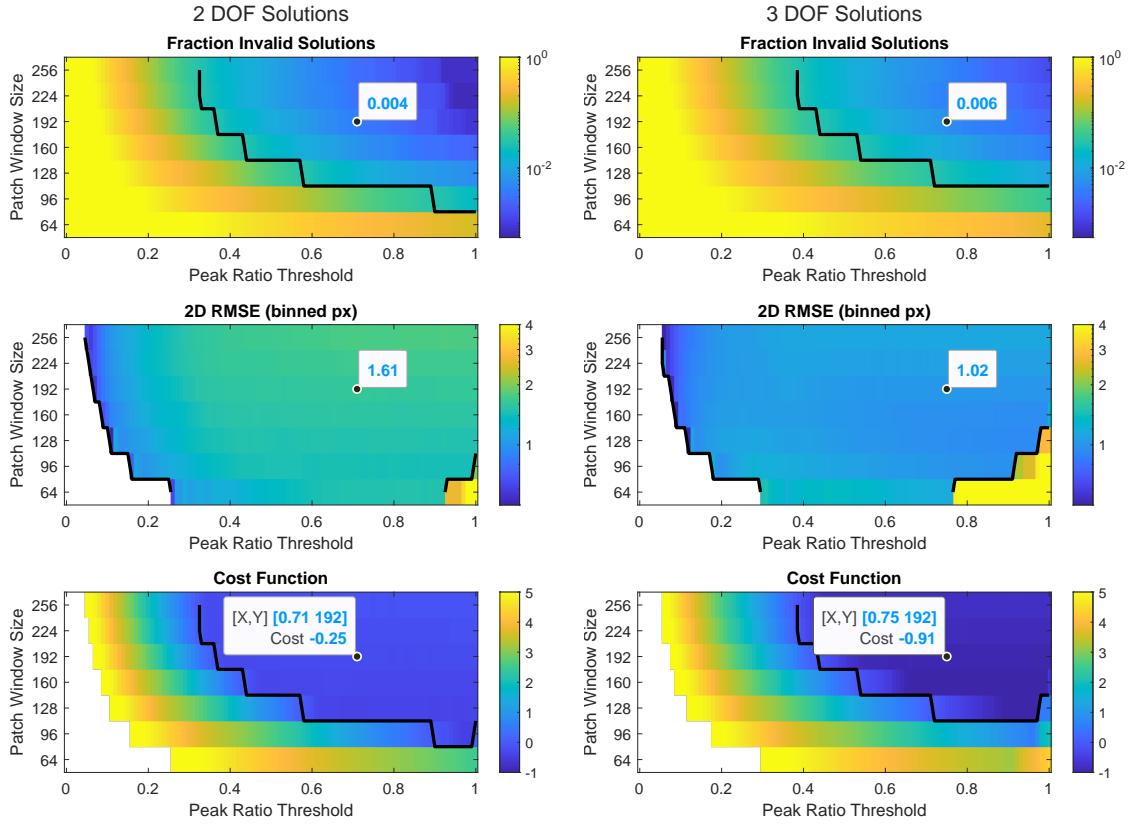
Performance at various window sizes was measured at various levels of bland terrain. Starting with a DTM of the Namib dunes and a filtered, flattened surface representing the base terrain under the dunes, the height of the dunes relative to the base terrain was scaled by various factors from 0 to 1. Images from a representative flight trajectory over these terrains were generated using the RCE software, with the flight modified such that the altitude the images were taken from were consistent across all terrain scale factors. Image pairs for velocimetry measurements were then selected across the entirety of the flight, and patch and full-image correlations were performed using states with knowledge error representative for velocimetry. Correlation measurements for each window size were then compared to truth, which was found by computing the center of the patch/full-image of the second image in the first (or reference) image and calculating the difference in pixel location. As shown in Figure 5, larger window sizes and higher terrain scale factors tend to yield an increase in the number of valid patches, where valid means the peak-ratio threshold parameter has been met. In contrast, the figure also shows that at larger terrain scale factors, smaller window sizes tend to have lower translation errors than larger window sizes. The right side of the families of curves show a sharp increase in translation error, where the peak ratio threshold has relaxed sufficiently to allow mis-registered correlations to pass.

While Figure 5 shows the performance of individual patches, multiple patches are correlated per image pair and can be combined in 2DOF and 3DOF batch solutions, with an increase in performance from outlier rejection. These results are shown in Figure 6.

Additionally, to select the operating parameters of patch window size and peak ratio threshold, an initial cost function was developed to optimize over the two dependent variables of fraction valid solutions and translation accuracy as shown in Eq. (15), where  $v$  is the fraction of valid solutions and  $\varepsilon_{tr}$  is the translation error.



**Figure 5 RMS 2D translation error vs fraction valid patches as parametric plots over peak ratio threshold for Tukey window sizes from 64 to 256 pixels across terrain scale factors from 0.2 to 1.0.**



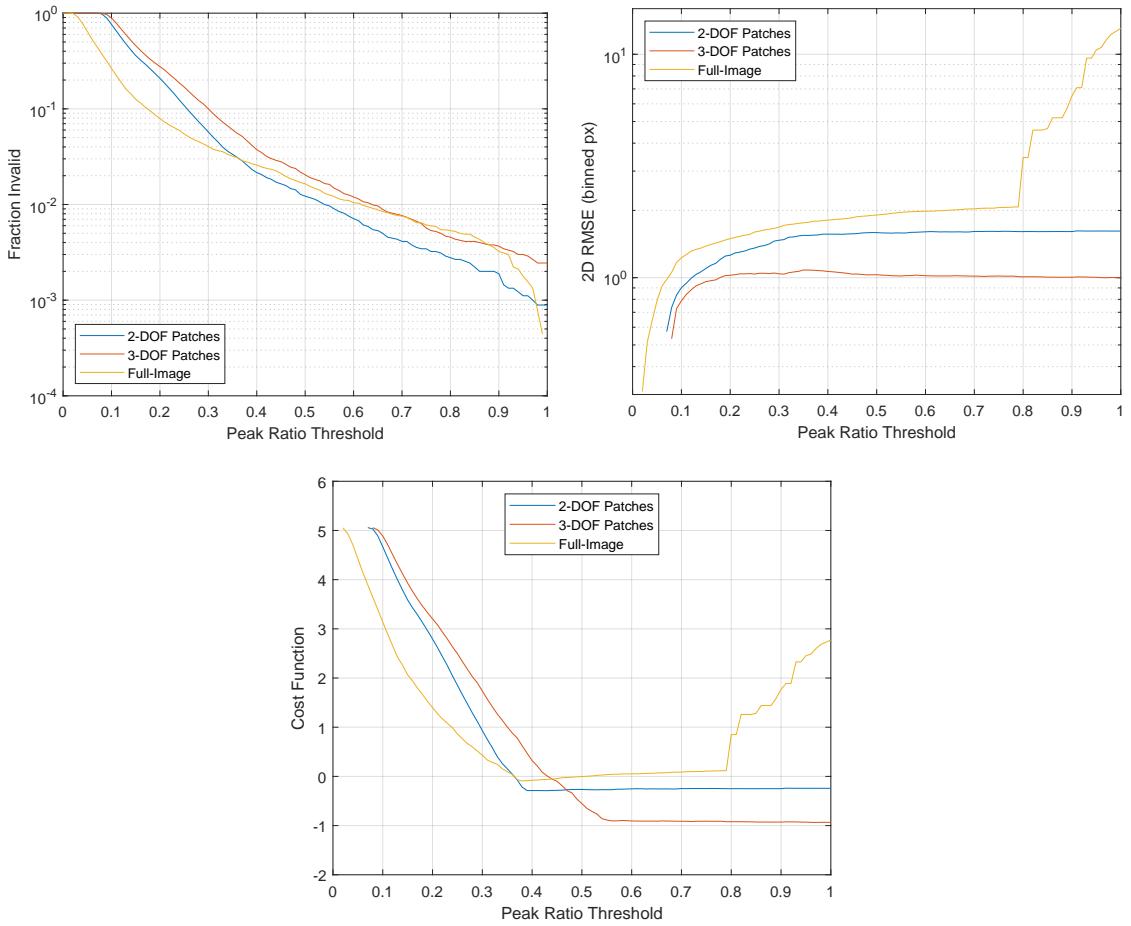
**Figure 6 2DOF and 3DOF batch solution performance as a function of window size and peak ratio threshold, averaged over terrain scale factors from 0.2 to 1 in steps of 0.2.** Top plot shows the fraction of invalid solutions, middle plot shows the RMS 2D translation error, and the bottom plot shows the cost function (Equation 15). The solid black line denotes the boundary between regions where the algorithm is meeting required performance and where it is not. Highlighted points are the selected optima, based on a centroid of the absolute value of the cost function within the region where the algorithm meets requirements.

$$cost = \max\left(\log_2 \frac{1-v}{1-0.97}, \log_2 \frac{\varepsilon_{tr}}{4.5 \text{ mrad}}\right) \quad (15)$$

The current cost function is designed such that a factor of 2x reduction in invalid solutions is weighted the same as a 2x reduction in translation error, with the worst performing metric determining the cost. Costs are normalized to the current performance requirements, with the requirements being met when the cost is negative.

Figure 6 shows the 2DOF solution in general having more valid solutions (due to only requiring two valid patches vs three for the 3DOF solution), while the 3DOF solution has a lower translation error. For the defined cost function, the 3DOF solution has the best performance, and both solutions indicate an optima at a window size of 192 and a peak ratio threshold in the 0.7 to 0.75 range.

Comparing the batch solution results against a full-image correlation (Figure 7) shows that a full-image correlation has the lowest rate of invalid measurements for peak ratio thresholds less than 0.36



**Figure 7 2DOF and 3DOF patch-based batch solution and full-image correlation performance, both as a function of peak ratio threshold, averaged over terrain scale factors from 0.2 to 1 in steps of 0.2. Patches use a Tukey window of size 192. Top-left plot shows the fraction of invalid solutions and top-right plot shows the RMS 2D translation error, and the bottom plot shows the cost function.**

and above 0.97, while the 2DOF solution performs better in the middle range. The 3DOF solution has the lowest translation error so long as it has any valid measurements (peak ratio threshold above 0.08). On balance, the cost function indicates that the 3DOF solution has the best performance overall, with performance nearly flat over peak ratio thresholds between 0.55 and 1.

### Integrated Test Platform Analysis

In addition to the image correlation performance using our software models shown above, the imagery obtained during flights of the Integrated Test Platform in the Imperial Dunes of California can be used to explore ETS robustness in a terrestrial environment. A ten minute flight from May 12, 2022 at 7:30 PM was selected as a representative flight due to the twilight illumination conditions precluding sharp shadows from providing unrealistic features in the imagery. An example image from the flight is shown, post-calibration, in Figure (8).

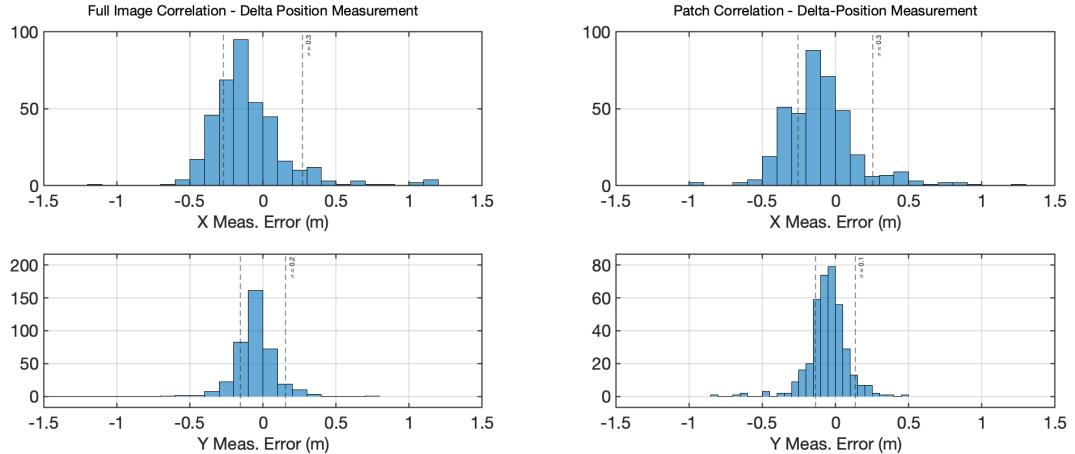
Unfortunately, due to the lack of a high-resolution DTM of the flight range in the Imperial Dunes,



**Figure 8** ITP flying over the Imperial Dunes, CA, and representative image, post-calibration, from a 7:30 PM flight of the ITP on May 12, 2022.

correlation accuracy statistics cannot be directly computed using these flight data. Instead, pseudo-measurements of delta position between two camera frames using the image correlation results are derived and compared to GPS knowledge. These pseudo-measurements are akin to the older version of ETS and the batch measurement variation.

Figure (9) shows the velocimetry delta-position psuedo-measurement errors from replaying the flight data, including the a priori navgiation errors from the flight, through an open-loop model to recompute ETS image correlation results.



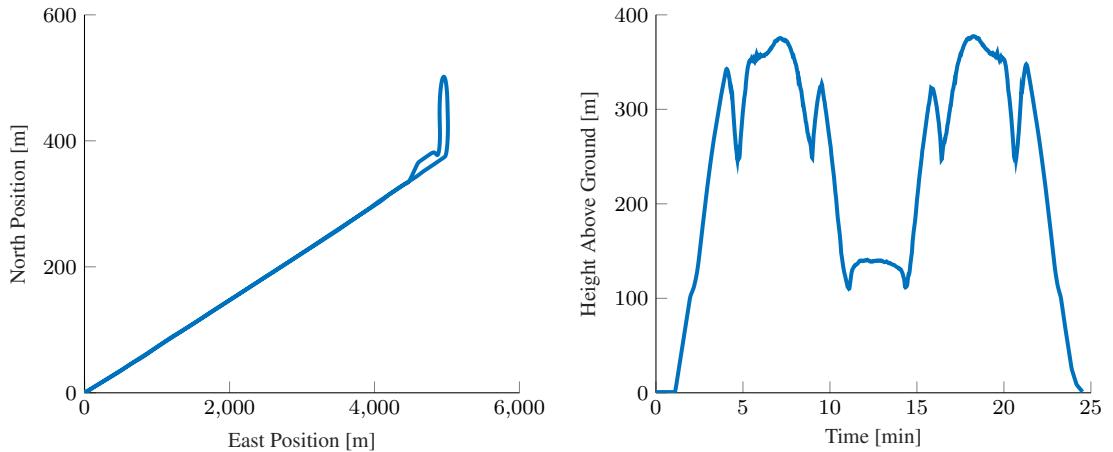
**Figure 9** Psuedo-measurement delta position errors using full image and patch image correlations from a 7:30 PM flight of the ITP on May 12, 2022.

### Linear Covariance Analysis

The linear covariance analysis (LCA or LinCov) serves as a standard approach for evaluating the performance of a linear filter (i.e. operates on a single mean and covariance), such as an Extended Kalman Filter (EKF), in real-time systems.<sup>11</sup> In essence, this method involves the propagation and updating of the filter covariance matrix along a specified trajectory. While it does not replace Monte Carlo analyses in closed-loop simulation environments, as it doesn't compare filter estimates to

simulated truth, it remains a valuable tool for quickly assessing different filter configurations and gaining insights into the impacts of various sensors and flight profiles.<sup>12</sup>

In this analysis, we focus on the “scout” flight depicted in Figure (10), where *Dragonfly* takes off to survey a potential landing site before returning to its take-off location. The LinCov results for this flight are presented in Figures (11)–(12), comparing 1 Hz velocimetry measurements using full-image correlation to those using individually filtered patch-correlation measurements. Notably, certain states, such as vertical position and velocity, exhibit consistent uncertainty regardless of the velocimetry method, as the knowledge in these states is driven mostly by the LiDAR and barometer measurements as described in 13, where the measurement noise standard deviations for these measurement types are 0.5 m and 12.5 Pa, respectively. However, improvements are observed in other uncertainties, particularly those related to lateral position, velocity, and accelerometer biases.



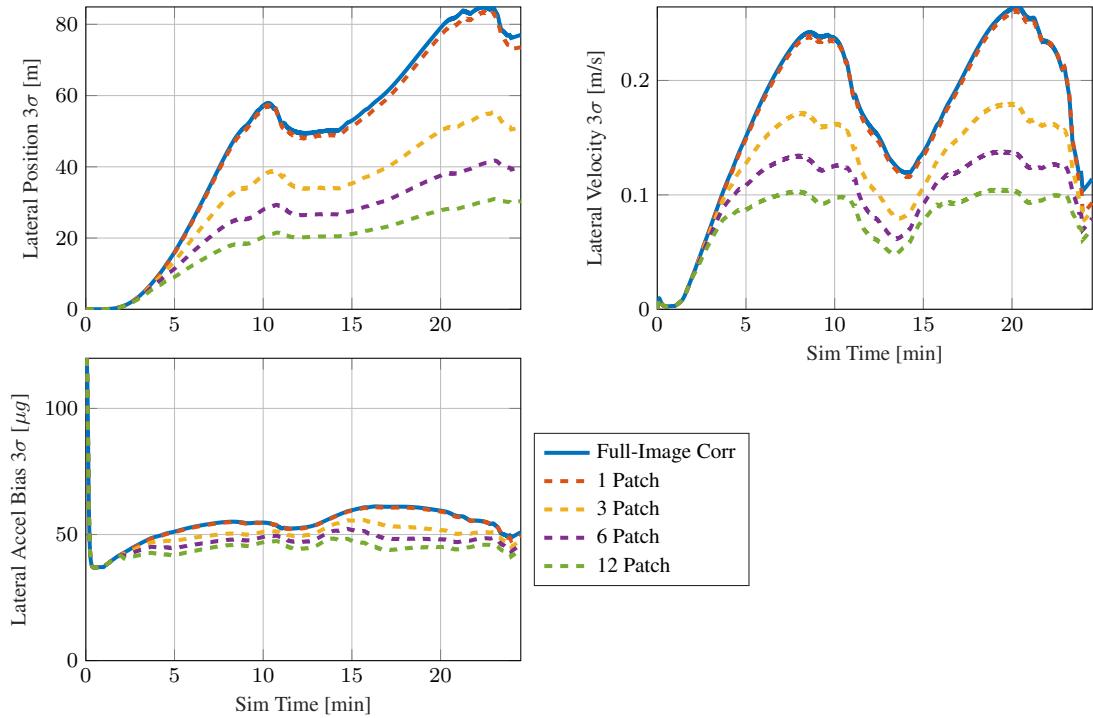
**Figure 10** Profile of scout flight for LinCov.

The results of Figure (11) show how different numbers of patches affect the filter estimate, as compared to the full-image correlation method. The lateral position and velocity uncertainties do not include contributions from heading error relative to the global Titan reference frame. The patch selection process for LinCov was selected to mimic the ETS procedure, such that one patch occurs near the center of image, as this is the point where the range-to-ground is well-known due to the significantly smaller LiDAR FOV relative to the NavCam FOV. The remaining patches are uniformly distributed throughout the image, but are spaced appropriately so they do not overlap. This spacing is enforced as each patch measurement is assumed uncorrelated with the others. Referring again to Figure (11), it is clear that a single, center patch performs almost identically to full-image correlation, with small differences caused by the small differences in the patch measurement model. Increasing the number of patches (3, 6, and 12) clearly decreases the lateral uncertainties significantly as the filter treats the patches as independent measurements.

Of course, the formulation of the measurement noise is key, and these LinCov analyses will need to be validated by future Monte Carlo analyses. As seen in Eq. (13), the measurement noise model for patches is

$$\boldsymbol{\eta} = \boldsymbol{\eta}_{\text{corr}} + \boldsymbol{\eta}_z, \quad (16)$$

where  $\boldsymbol{\eta}_{\text{corr}} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{corr}}^2 \mathbf{I}_{2 \times 2})$  is the patch correlation white noise reported by the ETS process,



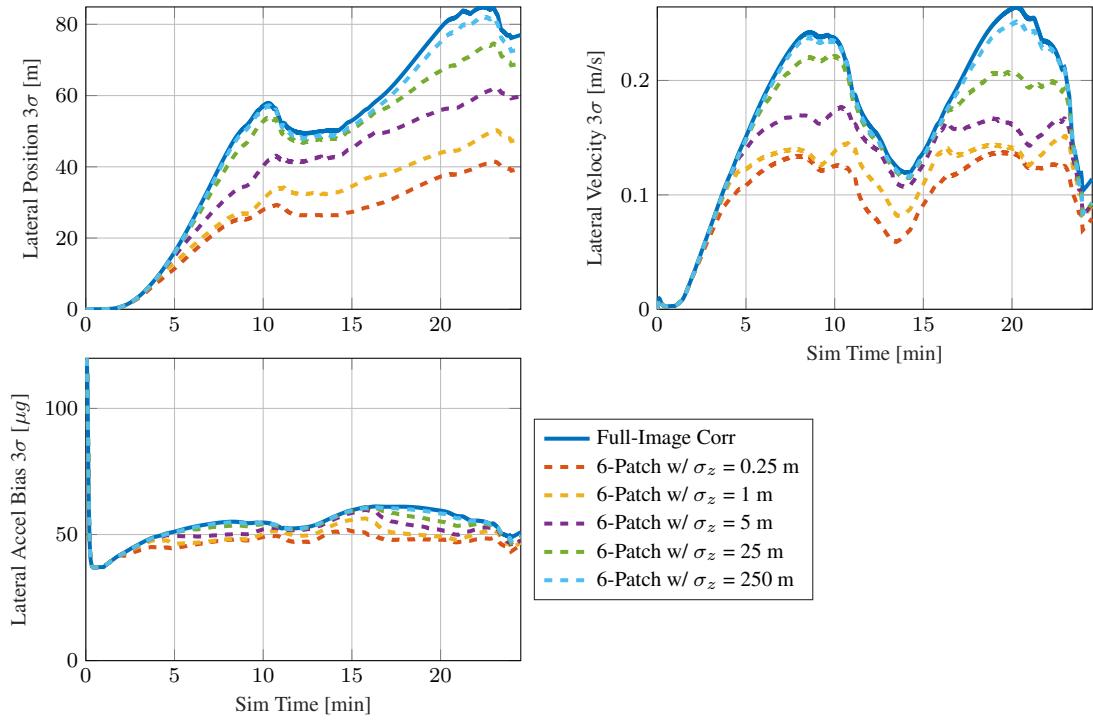
**Figure 11** Comparison of LinCov results for scout flight with full-image correlation and individually filtered patch velocimetry measurements, with various number of patches.

which for LinCov is selected to be  $\sigma_{\text{corr}} = 2$  pixels for both full-image correlation and patches. On the other hand,  $\eta_z \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_z)$  is noise attributed to the uncertainty in range from camera to surface at the center of a patch location. This is modeled as

$$\mathbf{R}_z = \mathbf{H}_z \sigma_z^2 \mathbf{H}_z^T, \text{ where } \mathbf{H}_z \approx \frac{1}{z_p(t_r)} \mathbf{A} \mathbf{R}_{\text{NED}}^{\text{CAM}}(t_r) \mathbf{R}_{\text{CAM}}^{\text{NED}}(t_c) \mathbf{A}^{-1} \begin{bmatrix} i_c \\ j_c \\ 1 \end{bmatrix}, \quad (17)$$

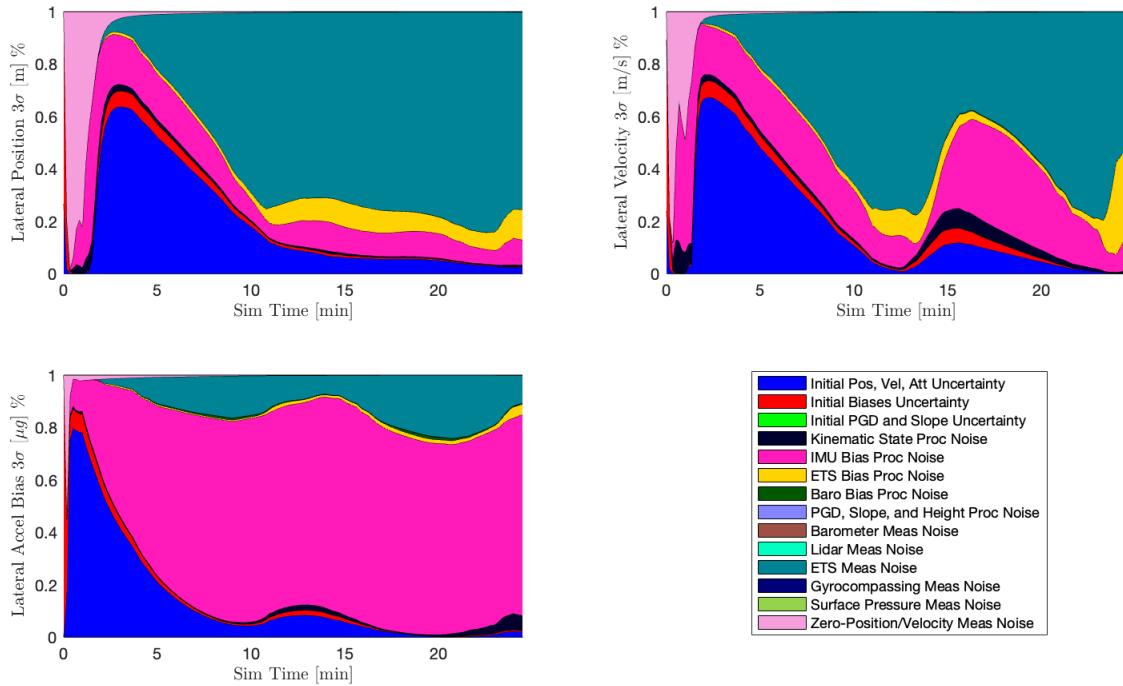
and where  $\sigma_z$  is the standard deviation in the estimate of  $z_p(t_c)$ , which is then mapped into the measurement space via  $\mathbf{H}_z$ . This is largely a function of the patch distance from the center of the LiDAR FOV. As *Dragonfly*'s navigation filter models the ground as a flat plate, this uncertainty becomes worse as patches move further from the center of the image, and is especially bad outside the LiDAR FOV. As such,  $\sigma_z$  will need to be tuned carefully during Monte Carlo analyses, but its effect on the filter covariance can be examined in Figure (12), where  $\eta_z$  is only added to patches outside the LiDAR FOV, and thus the center patch ensures that the worst possible patch performance is approximately bounded by the full-image correlation covariance. Examining Figure (12), it is clear that the patch velocimetry measurements are sensitive to the noise in  $\sigma_z$ , and thus an important parameter for future tuning.

For more insight into the improvements that patch measurements provide, an error budget analysis is employed—a frequent complement to LinCov due to its use of the same linear filtering components.<sup>14</sup> Essentially, an error budget allows filter designers to understand the contributions of different error sources to the overall uncertainty of specific states. The error budget results for full-image correlation velocimetry are illustrated in Figure (13), whereas the results for six indi-

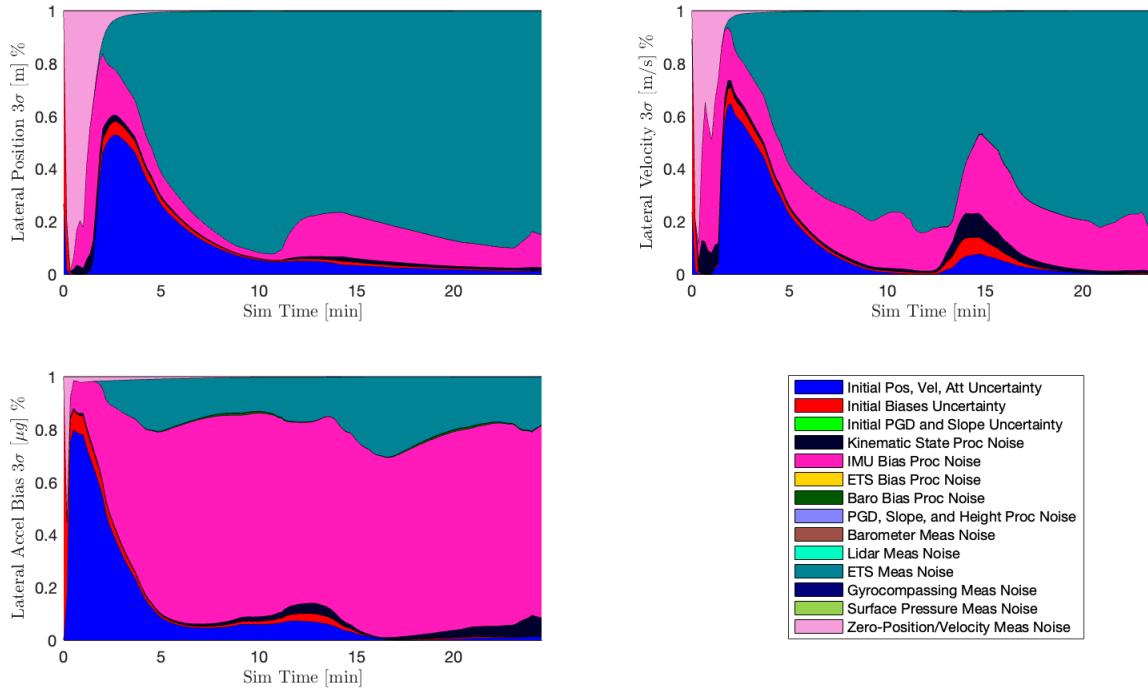


**Figure 12** Comparison of LinCov results for scout flight with full-image correlation and six individually filtered patch velocimetry measurements with varying  $\sigma_z$ .

ividually filtered patch measurements are found in Figure (14). The differences between these two plots are minimal, but it is worth mentioning that the full-image correlation model includes two bias states (modeled as First-Order Gauss Markov states in the  $x$  and  $y$  image plane),<sup>15</sup> whereas the patch measurement model does not.



**Figure 13** Error budget results for scout flight with full-image correlation velocimetry measurements.



**Figure 14** Error budget results for scout flight with six independently filtered patch velocimetry measurements.

## CONCLUSION

The ETS and Navigation members of Dragonfly’s Mobility team have recently advanced the image processing design to enable correlating several patches of the most recent image to a reference image. This capability can be applied to generate both velocimetry and breadcrumb measurements driven by patch correlations at 1 Hz. The preceding analysis clearly shows that in open-loop LinCov analysis, both image correlation performance metrics, and linear covariance analysis favor patch correlation measurements over a measurement generated by a single full-image correlation. In addition, it is expected that processing individual patch measurements in the navigation filter, rather than generating a batched measurement from the patch correlations themselves will lead to a more robust overall design. For example, it removes the sensitivity of using a priori navigation filter information in the generation of the ETS measurement. This is especially useful, as a future effort for ETS and the navigation team is to explore methods to operate in a degraded mode in the event of a LiDAR dropout. However, before making a final decision on the measurement formulation, the team plans to run all three methods through a Monte Carlo analysis with the Mobility closed loop simulation that models ETS and the navigation filter running in sync with *Dragonfly*’s flight dynamics.

## ACKNOWLEDGMENT

The authors would like to acknowledge the incredible work by many members of the Dragonfly Mobility team in generating analysis results, in delivering high quality environmental and camera models, and in producing an extremely efficient FPGA implementation: Arpit Amin, Nishant Mehta, Carolyn Sawyer, Ron Smith, Justin Thomas, Minh Quan Tran, and John Youssef.

## REFERENCES

- [1] J. W. Barnes, E. P. Turtle, M. G. Trainer, R. D. Lorenz, S. M. MacKenzie, W. B. Brinckerhoff, M. L. Cable, C. M. Ernst, C. Freissinet, K. P. Hand, and e. al., “Science goals and objectives for the Dragonfly Titan rotorcraft relocatable lander,” *The Planetary Science Journal*, Vol. 2, No. 4, 2021, p. 130.
- [2] R. D. Lorenz, *Saturn’s Moon Titan; Owners’ Workshop Manual*. Haynes, 2020.
- [3] T. G. McGee, P. E. Rosendall, A. Hill, W. J. Shyong, T. B. Criss, C. Reed, G. Chavers, M. Hannan, C. Epp, and M. Nishant, *APLNav: Development Status of an Onboard Passive Optical Terrain Relative Navigation System*.
- [4] I. R. Witte, D. L. Bekker, M. H. Chen, T. B. Criss, S. N. Jenkins, N. L. Mehta, C. A. Sawyer, J. A. Stipes, and J. R. Thomas, “No GPS? No Problem! Exploring the Dunes of Titan with Dragonfly Using Visual Odometry,” *AIAA Scitech 2019 Forum*, 2019, p. 1177.
- [5] C. Harris and M. Stephens, “A Combined Corner and Edge Detector,” *Alvey Vision Conference*, Vol. 15, Citeseer, 1988, pp. 10–5244.
- [6] D. C. Brown, “Decentering distortion of lenses,” *Photogrammetric engineering*, Vol. 32, No. 3, 1996, pp. 444–462.
- [7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: a Versatile and Accurate Monocular SLAM System,” *IEEE transactions on robotics*, Vol. 31, No. 5, 2015, pp. 1147–1163.
- [8] B. Schilling, B. Foust, K. Jinho, T. McGee, and C. Boss, “Initial Flight Test Performance of the Dragonfly Navigation Filter,” *AAS/AIAA Astrodynamics Specialists Conference*, 2023.
- [9] I. R. Witte, S. Jenkins, S. Bibelhauser, N. L. Mehta, and J. Thomas, “Preliminary Flight Testing of Dragonfly’s Electro-optical Terrain Sensing Function,” *3rd Space Imaging Workshop*, 2022.
- [10] C. D. Neish, R. D. Lorenz, R. L. Kirk, and L. C. Wye, “Radarclinometry of the sand seas of Africa’s Namibia and Saturn’s moon Titan,” *Icarus*, Vol. 208, No. 1, 2010, pp. 385–394.
- [11] D. K. Geller, “Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1404–1414.
- [12] C. D’Souza and R. Zanetti, “Navigation and Dispersion Analysis of the First Orion Exploration Mission,” *AAS/AIAA Space Flight Mechanics Meeting. Vail, CO*, 2015.
- [13] B. Schilling, T. McGee, R. Mitch, and R. Watson, “Preliminary Design of the Dragonfly Navigation Filter,” *AAS/AIAA Space Flight Mechanics Meeting*, 2023.

- [14] K. C. Ward, G. S. Fritsch, J. C. Helmuth, K. J. DeMars, and J. S. McCabe, “Design and Analysis of Descent-to-Landing Navigation Incorporating Terrain Effects,” *Journal of Spacecraft and Rockets*, Vol. 57, No. 2, 2020, pp. 261–277.
- [15] J. R. Carpenter and C. N. D’Souza, “Navigation filter best practices,” Tech. Rep. TP-2018-219822, NASA, 2018.