SSC25-P2-40

Development of the Optical Deep-Space Instrument for Navigation (ODIN)

Ava Thrasher, John Christian, William Driessen
Georgia Institute of Technology
756 W Peachtree St NW, Atlanta, GA 30308
avacthrasher@gatech.edu

Rebecca Inman, Riana Pecourt, Ronney Lovelace, Max Marshall
NASA Johnson Space Center
2101 E NASA Pkwy, Houston, TX 77058
rebecca.inman@nasa.gov

ABSTRACT

The Optical Deep-space Instrument for Navigation (ODIN) is a customizable, multiple camera, multiple field-of-view (FOV) software and hardware sensor package. The instrument and software design are intended to make optical navigation (OpNav) methods more widely available to small satellite missions. Towards this goal, ODIN was designed to tackle multiple barriers that smaller missions face when considering OpNav. The flight software, built in core Flight System (cFS), is largely agnostic to camera selection and placement, such that the base software can accommodate mission specific designs. The flight software also has many autonomous functionalities such as target determination, attitude estimation, and horizon-based position estimation, while still offering the option for manual modes to ease mission operation constraints. The general physical architecture assumed for ODIN is multiple fisheye cameras to monitor the majority of the celestial sphere and identify OpNav imaging targets, and at least two narrow angle cameras to collect images for simultaneous attitude and position estimation. To demonstrate the ODIN flight software and one potential ODIN instrument configuration at a reasonable size for a small satellite mission, a 2U prototype with two very wide-angle cameras and two narrow-angle cameras was designed, constructed, and tested on the ground. Currently, ODIN is in its second year of development under NASA's University SmallSat Technology Partnership (USTP) program and has already passed multiple testing benchmarks, with a final goal of advancing the technology to TRL 6 (validation in a relevant environment). This paper captures the work done over this period to develop ODIN, including the algorithms used, the design of the flight software, and the design of the prototype and its use in testing. Some preliminary results are discussed, as well as the lessons learned and plans for further testing.

INTRODUCTION

Optical navigation (OpNav) has seen many advancements in recent years, including demonstrations of autonomous horizon-based measurements of the Earth and Moon on Artemis I.[1] However, to date, OpNav systems are of bespoke design, requiring intensive operator interaction, and often utilize very narrow field of view cameras, thereby imparting significant pointing and operating constraints. This can often make the inclusion of OpNav difficult for smaller missions.

The patent-pending ODIN design aims to create an autonomous, self-sufficient, and portable OpNav instrument that is compatible with the small satellite form-factor and typical mission constraints. The basic concept of operations, illustrated in Fig. 1, begins with collecting all images from the ODIN cameras and attempting to identify an imaging target (a known celestial body) in the images.

By localizing the body center of the known celestial body, ODIN can provide the spacecraft with a desired pointing vector for OpNav measurement collection. Once the spacecraft has performed the requested slew maneuver, ODIN will then collect images from both narrow angle cameras, one of which is observing a starfield and one which is now observing the celestial body. These images are then processed by the onboard flight software to produce an attitude estimate (using star observations) and a position estimate (using horizon-based OpNav), along with the corresponding covariance. A brief discussion about star identification and horizon-based OpNav algorithms that are used are included in this paper for reference. The flight software developed for ODIN

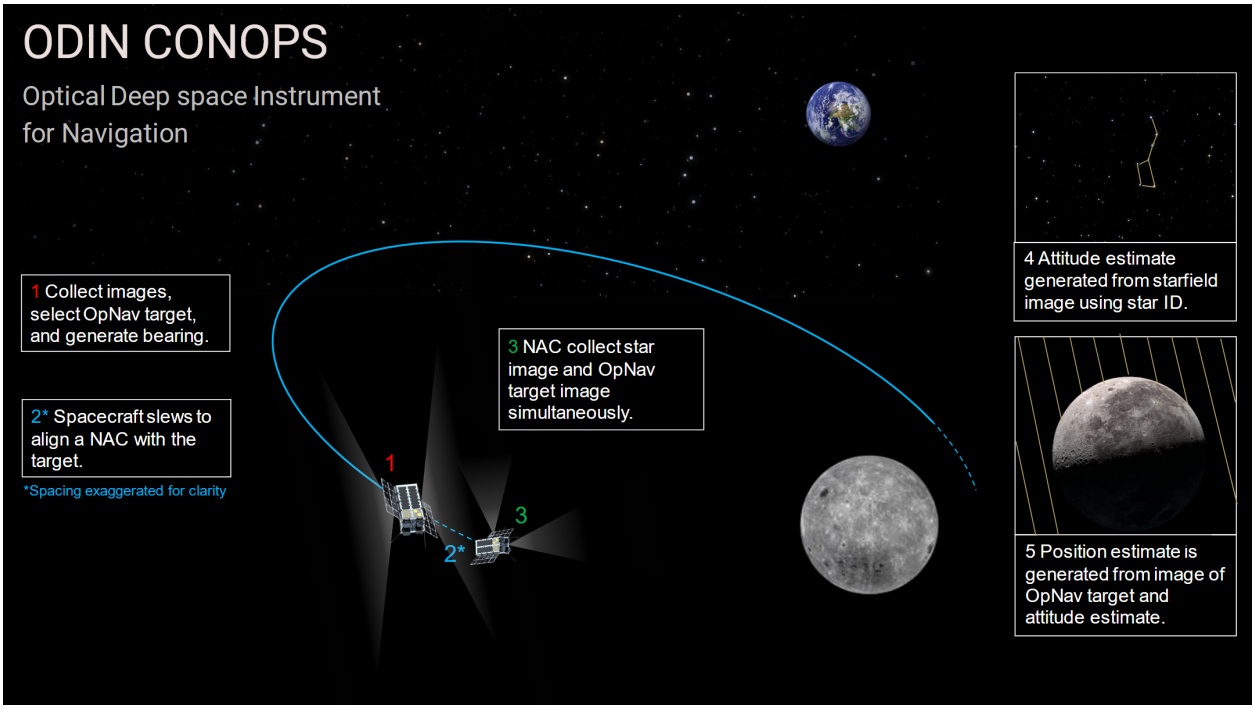

Figure 1: ODIN Concept of Operations

was built using core Flight System (cFS)[1] [2] and is designed as a base OpNav architecture which can interface with different numbers of cameras, different camera APIs, and different physical camera placements on the vehicle. This software uses many elements of the flight proven Artemis I OpNav flight code, with much of the ODIN follow-on work dedicated to enabling autonomous OpNav with multiple cameras. This paper further defines the interfaces between the user and ODIN, and the modes it can operate in.

This work also focuses on the methodology behind the hardware selections and testing. Most of the hardware needed for a full ground test has already been procured and tested as a stand in for flight hardware. This includes selections of camera sensors and lenses, flight computer, and other electronics. A significant amount of work shown here involved debugging interfaces between the camera, computer, and the ODIN application.

To further mature ODIN as a technology, future work includes a ground test of the autonomous functionality. Additionally, plans for further testing and usage for ODIN are discussed.

[1] https://github.com/nasa/cfe
[2] https://etd.gsfc.nasa.gov/capabilities/core-flight-system/

CAMERA MODELS

The ODIN architecture is intended to use a mixture of narrow and wide field-of-view cameras, including fisheye cameras. The camera models used to describe the projection of world points $[X, Y, Z]$ into pixel points (u, v) are described here.

NAC Projection and Distortion

Narrow angle cameras (NACs) are well modeled with the pinhole projection model as shown in Fig. 2. This model assumes a frame in which the Z-axis is aligned with the camera boresight and the X and Y axes are aligned with increasing pixel count, respectively.

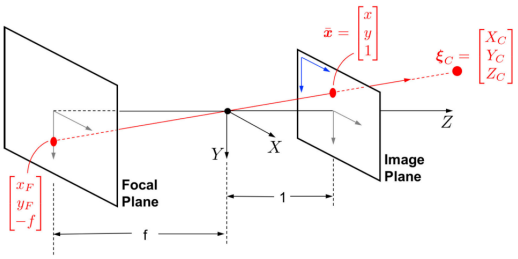

Figure 2: The Pinhole Projection Model is Used for Relatively Narrow FOV Cameras. Figure adapted from Ref. 2

In this projection, an incoming ray of light that passes through the optical center (red line in Fig. 2) is undistorted by the lens and in a straight line until it hits the focal plane. Rather than working with focal plane points, we chose to work with points on the image plane, which is a fictitious plane at unit length from the optical center along the Z-axis. The image plane point $\bar{\boldsymbol{x}}$ is related to the pixel point in the actual image $\boldsymbol{u}$ via the transformation

$$u = \frac{f}{\mu_x} x + \alpha y + u_p \tag{1}$$

$$v = \frac{f}{\mu_y} y + v_p \tag{2}$$

This transformation is a scaling by the focal length f converted to pixels via the pixel width μ_x and height μ_y, a coordinate shift from the principal point (u_p, v_p), and application of shear α. This transformation can be expressed nicely in a linear system of equation as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} d_x & \alpha & u_p \\ 0 & d_y & v_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boldsymbol{K}\bar{\boldsymbol{x}} \tag{3}$$

where $d_x = \mu_x/f$ and $d_y = \mu_y/f$. The resulting $\boldsymbol{K}$ matrix is often called the camera calibration matrix.

While NACs are well modeled by the pinhole projection, most have significant distortion in the image that must be accounted for in practice. The Brown-Conrady distortion model[3] is commonly used, and relates undistorted pixel points to distorted pixel points as

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots\right) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p1(r^2 + 2y^2) + 2p_2 xy \end{bmatrix} \tag{4}$$

WAC Projection and Distortion

Fisheye cameras do not follow the pinhole projection model. They instead are typically built to follow some other projection, with the most common ones being equidistance and equisolid.

To incorporate generically wide FOV cameras into ODIN, the parametric central camera model introduced by Scaramuzza et. al.[4] is used to relate how a image plane coordinate $\bar{\boldsymbol{x}}$ projects into the camera as a pixel point $\boldsymbol{u}$

$$\boldsymbol{g}(A\boldsymbol{u} + t) \propto \bar{\boldsymbol{x}} \tag{5}$$

In this model, the pixel coordinate is related to the image plane coordinate via an affine transformation defined by A and t, which transforms it to focal plane point $\boldsymbol{u}_f$ and the nonlinear function $g(u_f, v_f, f(u_f, v_f))$. For this implementation, the function f is

$$f(u_f, v_f) = a_0 + a_2\rho_f^2 + a_3\rho_f^3 + a_4\rho_f^4 \tag{6}$$

where ρ_f is the physical distance of the pixel from the optical axis.

Using this model, fisheye calibration involves estimating both the rotation and translation relative to a reference geometry (i.e. checkerboard) and the intrinsic parameters (a_0, a_2, a_3, a_4).

ALGORITHMS

Star Identification

Star identification (Star ID) is the primary algorithm that allows for on-board camera calibration and attitude estimation. For the narrow-angle cameras, ODIN will require two versions of star identification: for camera calibration and for the attitude estimation (when the camera calibration is known).

Both methods of identification rely on matching asterisms from the image to asterisms in a star catalog. In both cases, this is done by computing an invariant descriptor for the star patterns in the image and then comparing them to a catalog of star pattern invariants by querying some data structure.[5] The methods differ in how many stars are in the asterism, which invariants are used to describe them, and in this case, what data structure is used to store and access these invariants.

The number of stars needed for an identification scenario is dependent on how many independent invariants can be derived from the asterism, and whether that number of invariants is descriptive and unique enough to match. The number of

invariants G for a number of points d is related by

$$\mathrm{G} = 2d - k \tag{7}$$

where k is the degrees of freedom in the transformation.

When a camera is well calibrated, the stars in the catalog and the stars in the image can be related by only 3D rotation, or the action of a 3D special orthogonal group, $SO(3)$. This is a transformation with $k = 3$ degrees of freedom. This corresponds to $G = 3$ invariants. In practice, the interstar angles of star triads are used as asterism descriptors for calibrated cameras, but the reality of noisy measurements means that some additional stars must be verified after an initial match to ensure a it's unique. This triad asterism is shown in Fig. 3.

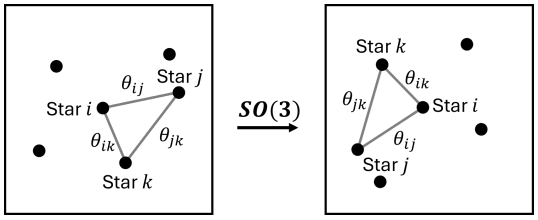

Figure 3: Inter-star Angles as Invariant Descriptor for a Well Calibrated Camera.

In the case of a narrow FOV camera for which the camera calibration is unknown or has degraded (due to launch conditions, temperature changes, etc.), the transformation between the stars in the image and the catalog can be described by a similarity transformation, which has $k = 4$ degrees of freedom. For this scenario, we select $d = 4$ stars for $G = 4$ invariants, since only 2 invariants will likely not be unique enough to find a correct initial match in a reasonable time. As before, this initial match of 4 stars must be verified with additional matches.

There are multiple choices for invariant descriptors for a quad of stars. For ODIN, inspiration was drawn from the Astrometry.net star identification.[6] In this approach, the stars from the image and catalog are represented in a common space defined by a similarity transform. The two stars farthest apart in the quad are labeled A and B, and transformed into a space where A is placed at the origin and B is placed at (1,1). The other two stars are labeled C and D and transformed to the same space as A and B. To differentiate between which star is C and which is D, C is always selected to be the star with the smaller X coordinate, and star C and star D are constrained to fall within the unit circle in the transformed space. The invariant descriptor for this quad of stars is then defined as the coordinates of stars C and D. The stars in the transformed space are shown in Fig. 4.

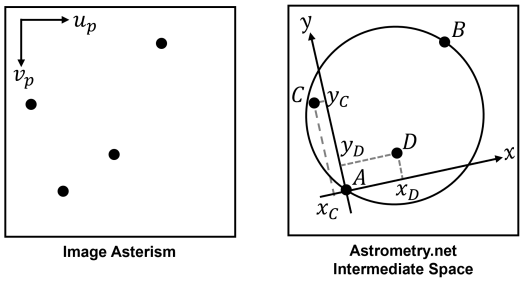

Figure 4: For an Uncalibrated, Narrow FOV Camera, Star Quads Can Form Descriptive Invariants for Star Identification.

Horizon-Based Position Estimation

The horizon-based position estimation used in ODIN uses the non-iterative Christian-Robinson algorithm,[2] which estimates the position of the observed celestial body relative to the observer given a set of points along the horizon (or limb) of the body.

Horizon points are extracted via a line scan in the direction of illumination. This line scan detects the pixel-level edge of the horizon, which can then be refined to a sub-pixel estimate using Zernike moment methods.[7] Using the calibrated camera matrix and accounting for distortion, we then transform the pixel coordinates of the horizon points to obtain n 3D vectors $\{\bar{\boldsymbol{x}}_i\}_{i=1}^n$ in the camera frame.

Large celestial bodies are well modeled as triaxial ellipsoids with a semi-major, semi-intermediate, and semi-minor axis of a, b, and c, respectively.[5] This allows us to define the shape matrix

$$\boldsymbol{A} = \begin{bmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/b^2 & 0 \\ 0 & 0 & 1/c^2 \end{bmatrix} \tag{8}$$

This shape matrix can be decomposed to the scaling matrix $\boldsymbol{D}$, which is the symmetric factorization of the shape matrix

$$\boldsymbol{D} = \begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/b & 0 \\ 0 & 0 & 1/c \end{bmatrix} \tag{9}$$

Using this scaling matrix, and the attitude from the camera frame to the principal axis frame $\boldsymbol{T}_C^P$, the vectors to the horizon points are transformed and normalized such that they point to a horizon point on a unit sphere

$$\bar{\boldsymbol{x}}_i' = \boldsymbol{D}\boldsymbol{T}_P^C\bar{\boldsymbol{x}}_i \tag{10}$$

$$\boldsymbol{w}'_i = \bar{\boldsymbol{x}}'_i / \|\bar{\boldsymbol{x}}'_i\| \tag{11}$$

The geometry then dictates the linear system of equations

$$\begin{bmatrix} \boldsymbol{w}_1'^T \\ \boldsymbol{w}_2'^T \\ \vdots \\ \boldsymbol{w}_n'^T \end{bmatrix} \boldsymbol{n}' = \mathbf{1}_{n\times 1} \tag{12}$$

Which can be solved for n' using least squares. This vector can then be transformed and scaled to obtain the estimate for the position of the celestial body relative to the observer in the observer frame

$$\boldsymbol{s}_C = (\boldsymbol{n}'^T \boldsymbol{n}' - 1)^{-1/2} \boldsymbol{T}_C^P \boldsymbol{D}^{-1} \boldsymbol{n}' \tag{13}$$

The covariance of this position estimation $\boldsymbol{P}_{\boldsymbol{s}_C}$ is also computed, using knowledge of the uncertainty in the horizon localization in the image. To do this, we begin by transforming the celestial body shape matrix $\boldsymbol{A}_P$ into the camera frame

$$\boldsymbol{A}_C = \boldsymbol{T}_C^P \boldsymbol{A}_P \boldsymbol{T}_P^C \tag{14}$$

and computing the matrix $\boldsymbol{M}_C$

$$\boldsymbol{M}_C = \boldsymbol{A}_C(\boldsymbol{s}_C \boldsymbol{s}_C^T)\boldsymbol{A}_C - (\boldsymbol{s}_C^T \boldsymbol{A}_C \boldsymbol{s}_C - 1)\boldsymbol{A}_C \tag{15}$$

which is proportional to the conic locus of the the horizon in the image plane.

Then for each horizon point the image plane coordinate covariance $\boldsymbol{R}_{\bar{\boldsymbol{x}}_i}$ is computed as

$$\boldsymbol{R}_{\bar{\boldsymbol{x}}_i} = (\sigma_{u_i}/d_x)^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{16}$$

where σ_{u_i} is the standard deviation of the pixel error in the horizon localization, and d_x is the focal length in pixels.

The variance of the residuals between the measured horizon and projected horizon conic $\sigma_{\epsilon_i}^2$ is then computed as

$$\sigma_{\epsilon_i}^2 = 4\bar{\boldsymbol{x}}_i^T \boldsymbol{M}_C \boldsymbol{R}_{\bar{\boldsymbol{x}}_i} \boldsymbol{M}_C \bar{\boldsymbol{x}}_i \tag{17}$$

The matrix $\boldsymbol{H}_i$ is computed as

$$\boldsymbol{H}_i = 2\boldsymbol{s}_C^T[(\bar{\boldsymbol{x}}_i^T \boldsymbol{A}_C \bar{\boldsymbol{x}}_i)\boldsymbol{A}_C - \boldsymbol{A}_C \bar{\boldsymbol{x}}_i \bar{\boldsymbol{x}}_i^T \boldsymbol{A}_C] \tag{18}$$

which was derived from and satisfies the first order approximation of the residuals between the measurements and the projected horizon ϵ_i as

$$\epsilon_i = \boldsymbol{H}_i \delta \boldsymbol{s}_C \tag{19}$$

Using these values for each horizon point, the inverse of the covariance for $\boldsymbol{s}_C$ is summed as

$$\boldsymbol{P}_{\boldsymbol{s}_C}^{-1} = \sum_{i=1}^{n} \frac{\boldsymbol{H}_i^T \boldsymbol{H}_i}{\sigma_{\epsilon_i}^2} \tag{20}$$

For additional details on the derivation of the covariance, please refer to Ref. 5.

OpNav Target Identification

Before attitude and position estimation can occur, one of ODIN's NACs must be pointed at an OpNav target, and the other NAC must be pointed at stars. To autonomously provide the vehicle information about where to point, ODIN will offer a target determination mode in which images from all ODIN cameras are captured and used to search for an OpNav target in the sky. This algorithm would produce a bearing measurement to the OpNav target, which would inform the desired relative attitude to align a NAC with the target for imaging. It is important to note that the resulting measurement of this algorithm is *not* used for state estimation, but instead provides the vehicle with information about where it should point to acquire OpNav measurements.

Using the images from ODIN's photometrically calibrated cameras, we can detect bright bodies in the image via a threshold (informed by the cameras background noise), which will result in n connected components in the image. Assuming ODIN is in the Earth-Moon system, there are only three bright extended bodies: Earth, Moon, and Sun.

When viewing the Earth and Moon from within the Earth-Moon system, either body may appear resolved or unresolved, depending on the distance. See Fig. 5.

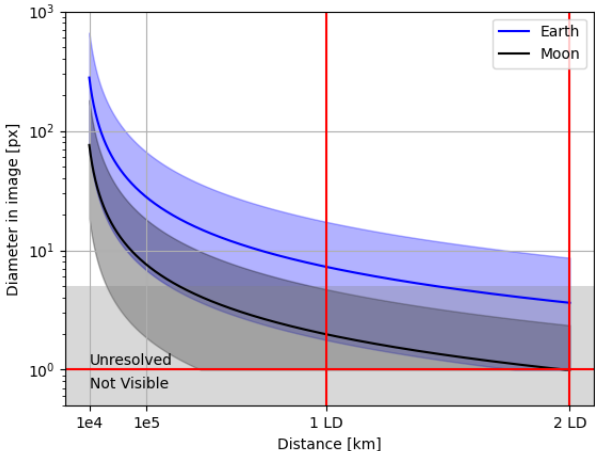

Figure 5: Pixel Diameter of the Earth and Moon for Current Hardware Selection. Colored Area Shows Maximum (Center of Image) and Minimum Diameter (Edge of Image) Possible.

The production of the bearing measurement to each unresolved detected body is straightforward given the camera projection model and the center of the detected celestial body in the image. The direction of illumination in the image is known from ephemeris data and the camera's attitude. Therefore, the center of the unresolved body can be estimated accounting for a photocenter offset caused by the phase angle.[8] In the case of a resolved body, more work will need to be done to estimate the center of the body by fitting an ellipse to its detected horizon points.

The Moon and the Earth have very different albedoes. Thus, the problem of differentiation between the Earth and the Moon may be solved using disk-integrated photometry. ODIN accomplishes this by computing the expected brightness for the Moon at the given sizes and phase angles g detected in the image and then comparing this to the image observations. It is important to account for the phase angle as shown in Fig. 6, as the brightness will change significantly depending on the illumination condition.

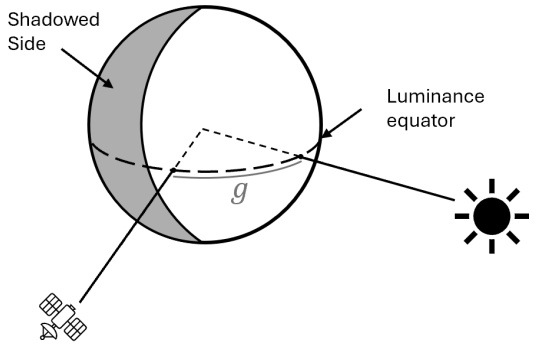

Figure 6: Expected Intensity of the Celestial Body Changes with Phase Angle (g).

The expected intensity (I) in units of W/Sr of the Moon will be modeled with the disk-integrated intensity of a Chandrasekar sphere as a function of the phase angle g. For brevity, the authors direct the reader to the full derivation of this function in Ref. 5.

The intensity estimated can be related to an irradiance E captured by the camera (so long as the entire body is captured in the frame)

$$E = I\Omega = I\frac{A}{r^2} \tag{21}$$

where A is the area of the camera aperture and r is the distance from the source to the camera. The digital number J produced in the camera from the unresolved body is related to irradiance by the integral

$$J = \frac{\alpha A \cos\phi\tau}{hc}\int_{\lambda_{min}}^{\lambda_{max}} QE_\lambda\left(\lambda E_\lambda\right)\delta\lambda \tag{22}$$

In the above equation, the digital number is dependent on camera parameters such as the analog gain (α), aperture area (A), the exposure time (τ), the camera pass-band ($\lambda_{min}, \lambda_{max}$), and the spectral quantum efficiency of the sensor (QE_λ). It is also dependent on incident angle between the camera boresight and the captured object (ϕ), and the spectral irradiance of the unresolved body (E_λ). The constants h and c are Planck's constant and the speed of light, respectively.

In the absence of spectral information, average values can be used for the quantum efficiency and irradiance over the camera pass-band. This integral can then be approximated as

$$\bar{J} = \alpha\bar{Q}\bar{E}\left(\frac{\lambda_{max}^2 - \lambda_{min}^2}{2hc}\bar{E}\right)A\cos\phi\tau \tag{23}$$

where barred values are taken as the average.

The digital number value as computed as in Eq. (23), represents the summation of all of the pixel values across the detected disk or crescent. Given this, the comparison to be made is the summed values of all of the detected body pixels, and the compute digital number from the disk-integrated photometry. If the summed digital numbers of one of the detected bodies in the image uniquely matches the expected value to within the uncertainty of our calculation, then that body is assigned as the Moon. Once the Moon (or Earth) has been identified, ODIN will produce a desired bearing for the spacecraft which would align one of the NACs with the Moon (or Earth).

FLIGHT SOFTWARE ARCHITECTURE

The ODIN flight software package is written as a C++ cFS application. At a high level, the software is designed to

1. Receive and process commands and input data.
2. Perform camera calibration.
3. Perform target determination using WACs.
4. Take simultaneous images from NACs.
5. Produce a position and attitude measurement from NAC images.
6. Output measurement and diagnostic data.

The ODIN system is a sensor package, not a navigation solution. As such, ODIN produces a position and attitude measurement in the ODIN-defined frame.

The ODIN application has 4 modes: *IDLE*, *CALIBRATION*, *TARGET DETERMINATION*, and *PROCESSING*. In *IDLE* mode, the ODIN application is responsive but not taking images or performing and processing. In *CALIBRATION*, ODIN will take a series of NAC images to estimate the camera focal length and distortion characteristics, as well as compute the interlock angle between the cameras. In *TARGET DETERMINATION* mode, ODIN will take images from the WACs, identify the target(s) in the images, and compute a pointing vector that will put one of the targets in the field of view of one of the NACs. This provides a "lost-in-space" functionality that can assist the vehicle in finding valid targets. Lastly, in *PROCESSING* mode, ODIN will take simultaneous images from the NACs at the user commanded rate and process the images to produce a position and attitude measurement. ODIN can utilize the WACs and the target identification functionality to independently determine what target is in the field of view, eliminating the need for operator interaction to provide that information. However, the use of the WACs is not strictly necessary. If a mission design has no need for lost-in-space functionality, or the vehicle design is such that wide angle cameras are not feasible, the position and attitude estimation functionality is still usable. ODIN can accept an external command that specifies the camera/target configuration.

Building Blocks of ODIN FSW

The end-product for the ODIN project is a cFS application; however, the intention is to have a "core" product that is framework AND camera agnostic. The "core" software will define a generic interface to configure (both application and cameras), run the main sensor process, and handle commands, telemetry, and messages.

Camera Agnostic Interfacing

One of the main goals of the ODIN application design is to be as camera agnostic as possible. Previous OpNav systems have the camera control software tightly intertwined in the operations of the system, meaning changing out the camera hardware often requires significant and invasive code changes. While there will always need to be some level of custom code to interface with the specific camera hardware, ODIN aims to abstract the camera control code such that, if the user can provide the interface to the camera and implement a predefined list of necessary functions, then theoretically any camera/API can be used and no other changes to the application are required.

Camera Triggers and Timetag Syncing

In order to utilize the dual camera setup to get attitude from one image and position from the other, the images must be taken at the same time. To be clearer, the center of integration (exposure) time must be the same. To do this, ODIN utilizes both a software and hardware trigger, where one camera is triggered via a command from the ODIN app and the other camera is triggered via the connection between the two cameras. The camera with the longest exposure time is triggered via software (designated the primary camera and the other the secondary). A trigger delay is computed based off the difference in exposure time between the cameras (or a delay of

zero if they have the same exposure), which will delay the trigger signal from the primary camera. This ensures that the center of integration time matches and the timetags of the images are synced. Lab testing shows that the timetags are synced to within less than a millisecond.

The physical hardware trigger was set up such that choice of hardware/software triggers for each camera can be determined at time of image request. This allows greater flexibility so that either camera could be used to capture starfields or Moon/Earth images rather than requiring each camera to be designated a target.

PROTOTYPE DESIGN

A physical test unit for the ODIN software is crucial to advancing the technology to TRL 5. The current prototype (as shown in Fig. 7) was designed to be a platform for running early versions of ODIN. This prototype is used to test data and commanding interfaces, electrical and hardware interfaces, and algorithmic performance.

Figure 7: ODIN v1 Prototype Hardware Test Unit.

Camera Design

The primary hardware selections for the ODIN cameras were the sensors and lenses. Ideally, both the narrow-angle cameras (NACs) and the wide-angle cameras (WACs) are small form factor with highly customizable interfacing and commanding. To meet this goal for this prototype, we chose the commercial-off-the-shelf (COTS) Teledyne Blackfly board level sensor (shown in Fig. 8) for both cameras, which is available in a variety of configurations. This COTS, small, lightweight sensor is commanded via a camera API, which is highly flexible in camera settings, commands, and data transfer. The Blackfly boards have options for monochromatic CMOS sensors, which is the better choice for optical navigation images (as compared to color, or CCD sensors). The design for the NACs and WACs diverge from this choice since each have different purposes, and therefore require different versions of the sensor and different lens hardware. A summary of key camera parameters are provided in Table 1.

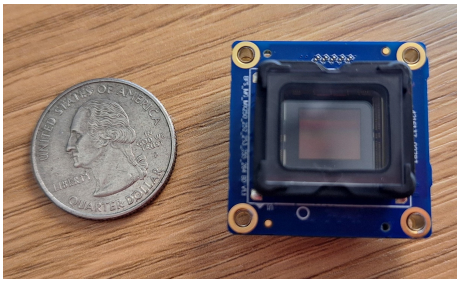

Figure 8: The Blackfly Board Level Sensor is a 29mm × 29mm × 10mm Board, Making it a Good SWAP Choice.

Table 1: Camera Specifications

Parameter	Narrow Angle Camera	Wide Angle Camera
Sensor	Sony IMX264	Sony IMX265
Resolution	2448×2048	2048×1536
Field of View (deg)	19	165
IFOV (arcsec)	28	387
Pixel Size (μm)	3.45	
Bit Depth Options (bits) (μm)	8, 12, 16	
Focal Length (mm)	25	1.8
Aperture Diameter (mm)	17.85 - 1.56	1.45
Total Mass (g)	59	19.6
Projection Model	Pinhole	Equidistance
Maximum Power Draw (W)	5	

The NACs are primarily used to capture images of stars to estimate camera attitude and images of a celestial body to estimate position. Therefore, careful consideration went into selecting a lens and sensor combination that had an instantaneous field of view (IFOV) capable of well-resolving the Moon in an image taken from the Earth. This involved striking a balance between resolution and FOV. Multiple sensor and lenses were considered, with each possi-

ble combination of sensor and lens indicated by the black asterisks shown in Fig. 9.

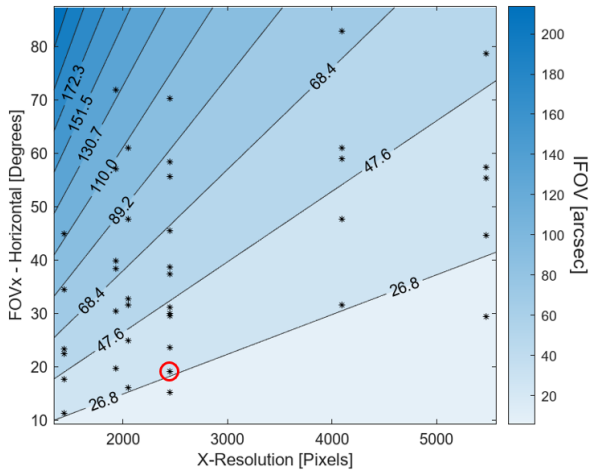

Figure 9: Trade Study for NAC Hardware. Black Asterisks Indicate a Sensor and Lens Combination. The Final Selection is Circled in Red.

The final selection is indicated in the red circle. This selection consisted of a 5MP Blackfly board level sensor and the Edmund Optics 25mm C-Series fixed focal length lens. This camera is well modeled by the pinhole camera model and Brown-Conrady distortion discussed in the background section.

Since the NAC cameras are used for high quality OpNav imaging, a custom light baffle was designed using the procedure from Ref. 5. This light baffle consists of four optimally-placed internal vanes to achieve a sun exclusion angle of 45 degrees. The prototype uses 3D printed baffle to test fit and function.

The WACs are used for finding bearings to imaging targets. For this purpose, the targets do not need to be resolved in the image. Instead, we prioritize a large FOV to maximize sky coverage. To achieve this, the 3.2MP Blackfly board level sensor and the Commonland Optics CIL239 fisheye lens were selected. With two of these WACs mounted on opposite ends of the prototype, they alone provide $\sim 87\%$ sky coverage. It is important to recall that the WAC is not well modeled as a pinhole camera. Instead, this camera is modeled with the Scaramuzza calibration[4] as discussed in the camera model section.

Flight Computer Considerations

The current ODIN prototype configuration uses a Raspberry Pi 5 for camera commanding as a turnkey solution to testing the camera selections. However, there are multiple flight computers being considered for the final prototype of ODIN, including the UP Board and LattePanda Delta 3. These boards were chosen for their potential ability to interface with the Blackfly board level sensors as well as their Single Board Computer (SBC) form factor.

An additional consideration is the physical cable interface between the flight computer and cameras. The board level sensors have an option for either a gigabit ethernet (Gig-E) connection or a USB 3 connection. At present, the Raspberry Pi 5 has been focused on due to its lower power requirements, but it lacks the ability to interface with the Gig-E cameras due to its incompatibility with Jumbo Ethernet Packets (MTU 9000).

Chassis Design

The current ODIN v1 prototype presented here is one potential physical manifestation of ODIN. In fact, the software developed for ODIN is being built to be agnostic to camera placement and type on the spacecraft. However, to demonstrate compatibility with small missions, this prototype was designed to fit into a 2U satellite form factor (10 cm $\times$ 10 cm $\times$ 20 cm). The ODIN team is considering numerous other physical manifestations not pictured here.

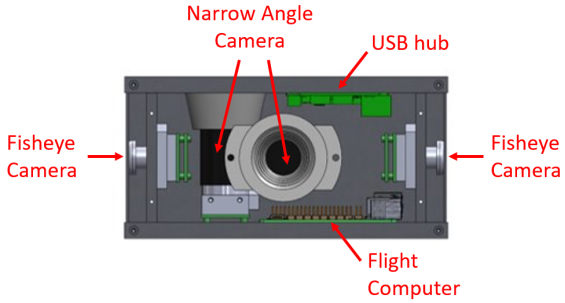

Figure 10: Interior View of ODIN v1 Prototype with Hardware Placement.

The most critical structure in ODIN v1 is the rigid optical base, which is currently a custom aluminum part connecting the two narrow angle cameras. In future testing, change in the interlock angle between the two NACs should be minimized to the highest degree possible via this optical base. Different materials more resistant to thermal deformation are also under consideration.

ODIN TESTING

At the project phase of ODIN that this paper captures, all hardware testing was carried out with ODIN version 0, as shown in Fig. 11. This preliminary test set-up allowed for camera calibration,

development of the hardware trigger connecting the cameras, testing the stability of the NAC interlock angle, and a demonstration of the pose estimation algorithms. However, this test set-up lacks many features of the previously presented prototype ODIN version 1. In this early version, the WACs are not integrated into the test unit, the NACs are not rigidly mounted on a common optical base, and the electronics interfacing is in a very preliminary design state. Additionally, the various pieces have been tested individually but not yet integrated into the ODIN application. Despite this, these initial results still yielded important insight into ODIN algorithms and provided critical information about design choices for the ODIN prototype.

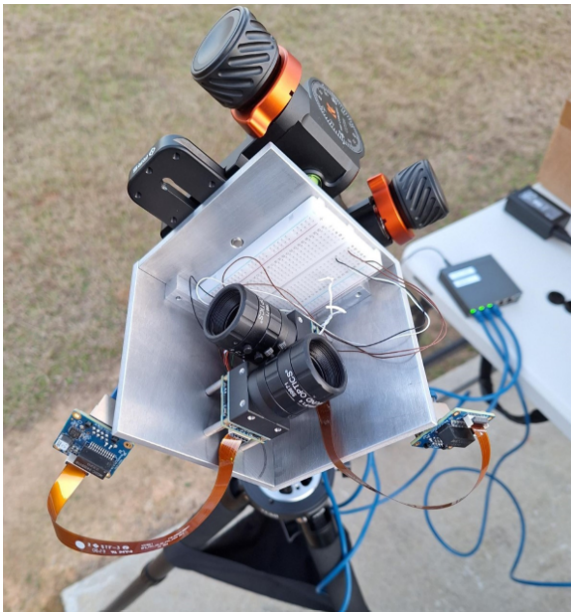

Figure 11: ODIN v0: Preliminary Test Set-up for Camera Calibration, Interlock Angle Experiment, and Pose Estimation Demonstration.

Camera Calibration

In order to use cameras with ODIN to produce a bearing measurement (from the WACs) and attitude and position estimate (from the NACs), the cameras must be geometrically calibrated.

The WACs are only used to produce a desired pointing information for the vehicle to begin OpNav, and are never used to produce any state estimate. Furthermore, this desired pointing vector only needs to be accurate enough to get the OpNav target in frame of the NAC. Therefore, it is sufficient to calibrate the WACs on the ground and use the derived projection model throughout operations.

The prototype WACs were calibrated with the checkerboard method detailed in Scaramuzza et. al.,[4] using the images shown in Fig. 12. The derived values are shown in Table 2.

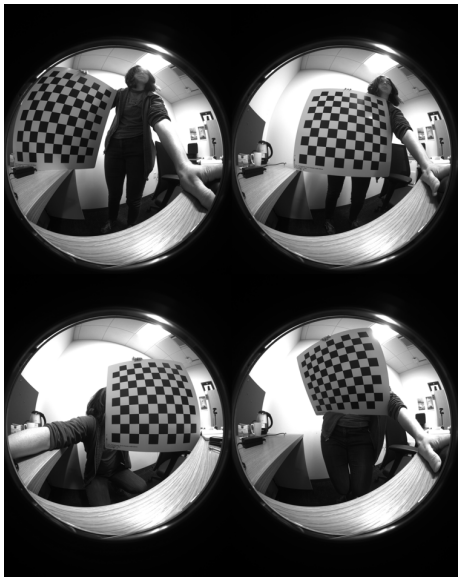

Figure 12: Mosiac of Fisheye Images Captured for Calibration.

Table 2: Fisheye Camera Calibration Parameters

Parameter	Value
a_0	533.039
a_2	-7.764e-04
a_3	4.340e-07
a_4	-8.639e-10

In contrast, the horizon points and the star centroiding from the NAC must be accurate to the subpixel level, meaning that calibration must be done on-board with stars to account for any changes that occurred during launch or any thermal effects on the camera structure in space.

To demonstrate this process, the ODIN NACs were calibrated using captured starfield images. For each camera, star detections across 15 images were identified using the uncalibrated star identification approach discussed previously. An example of one such image captured of the Orion constellation is shown in Fig. 13. Using only information about the camera resolution, this approach estimated the full camera calibration matrix and Brown-Conrady distortion parameters. These estimated parameters are shown in Table 3.

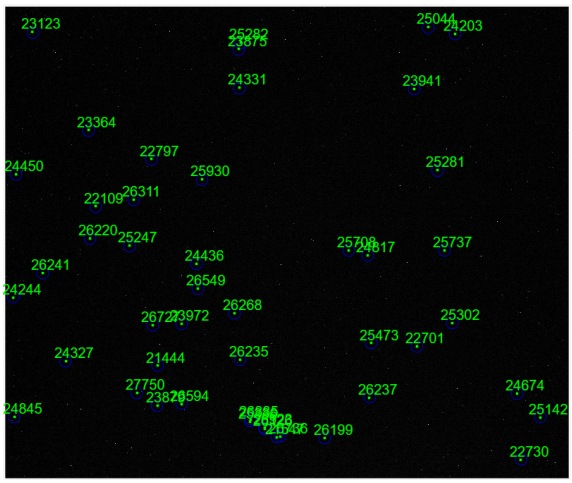

Figure 13: Uncalibrated Star Identification on ODIN NAC Image. Green Indicates Projected Stars with Hipparcos ID Label. Blue Circles Indicate Detected Star Centroids.

Table 3: Narrow Angle Camera Calibration Parameters

Parameter	Value
$f(mm)$	25.13
k_1	4.260e-02
p_1	-7.312e-04
p_2	3.857e-04

Hardware Trigger Development

A a critical functionality of cameras used for ODIN is the ability to simultaneously capture images at the same time from different cameras. This ensures that an image of a starfield from one camera (used for attitude estimation) is captured at the same time as the image of the celestial body from another camera (used for position estimation). To ensure precision with simultaneous image capture, the ODIN design uses a hardware trigger to avoid potential software latency. An example of our initial test set-up between two ODIN cameras is shown in Fig. 14.

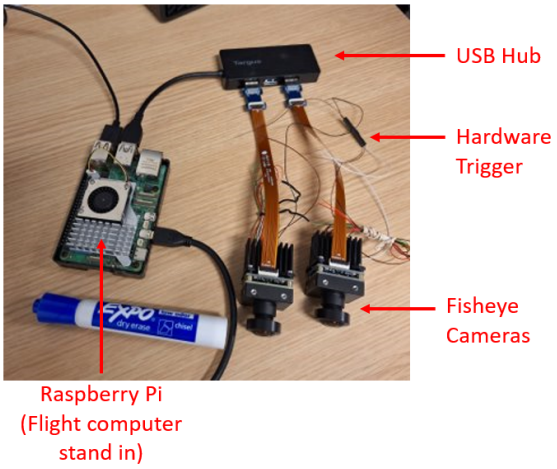

Figure 14: Hardware Trigger Testing Set-up.

This hardware trigger was tested by simply aiming both cameras at a clock and running a test in the ODIN flight software environment to capture an image with camera 1, which uses the physical cable connection to trigger camera 2. An example of two of these images taken simultaneously to validate the trigger's performance is shown in Fig. 15. Testing has shown that the hardware trigger is capable of consistently achieving simultaneous capture to the microsecond level.

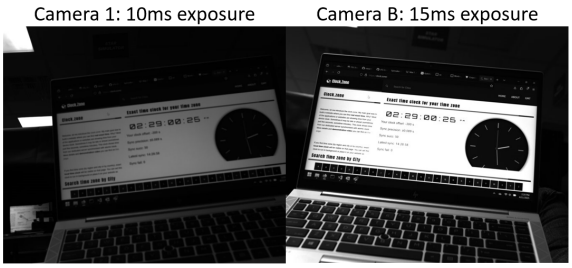

Figure 15: Simultaneous Images Captured with the Hardware Trigger.

NAC Interlock Angle

Accurately relating the attitude of one ODIN camera to another ODIN camera is another essential capability for the pose estimation algorithm to work correctly. Despite the fact that ODIN v0 did not have its two camera's rigidly connected via a common optical base, it was still important to establish a method for estimating the relative attitude between the two (i.e. the interlock angle).

To test this ability, simultaneous images of starfields from both cameras were captured using the test set up in Fig. 11, in which the cameras were mounted at ∼90 degrees apart. Calibrated star ID

was done for both images, producing an attitude estimate for both relative to the inertial frame. Over the course of the test, the test unit was reoriented a couple of times. The results of this test are shown in Fig. 16

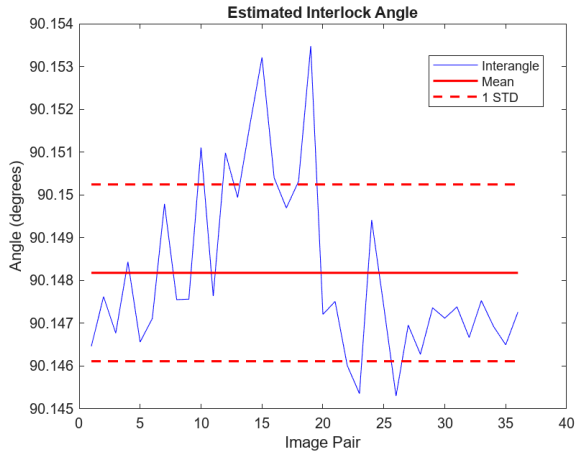

Figure 16: Magnitude of the Estimated Interlock Angle for ODIN v0 Over 36 Image Pairs.

The standard deviation of the estimated interlock angle over the course of the test was ∼7.2 arcseconds. This level of shifting could have been due to shifting or sag in the mechanical components of the test unit. Given that the IFOV of the NACs is 28 arcseconds, this level of shift in the camera could cause around pixel level of shifting to within 3σ. This is addressed in the ODIN v1 prototype with a rigid, common optical base, which will reduce any mechanical shifting.

Simultaneous Attitude and Position Estimation

The primary functionality of ODIN as an OpNav sensor is to produce a position estimate of the camera at the time of image capture. To do this, one NAC is aimed at a starfield and the other is aimed at the targeted celestial body. The images were then captured simultaneously, and the attitude from the star-field image was used to compute the attitude for the celestial body image. The position of the camera which captured the image of the celestial body can then be estimated.

To test this, the ODIN v0 testing unit was used to capture star and Moon images. A total of 21 image pairs were captured and processed from this field test. The estimated position of the Moon was then compared to the expected reference position as dictated by CSPICE ephemeris files.[9,10] Fig. 17 shows the estimated position and CSPICE reference position of the Moon projected into the image using the estimated attitude and camera model.

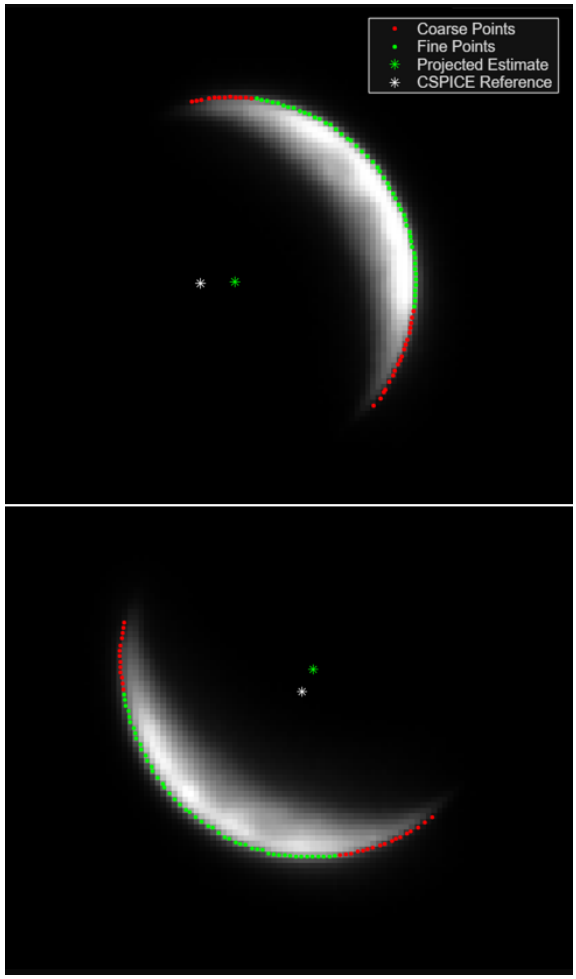

Figure 17: Zoomed in ODIN Images of the Moon with CSPICE Reference and Estimated Moon Position Projection.

In all of the images, there is a clear difference in the reference projection as compared to the estimated position, which varies between 4 and 7 pixels depending on the image, as shown in Fig. 18. A few possible sources for this discrepancy are currently being considered, including atmospheric refraction, and change in interlock angle estimation due to mechanical shifting.

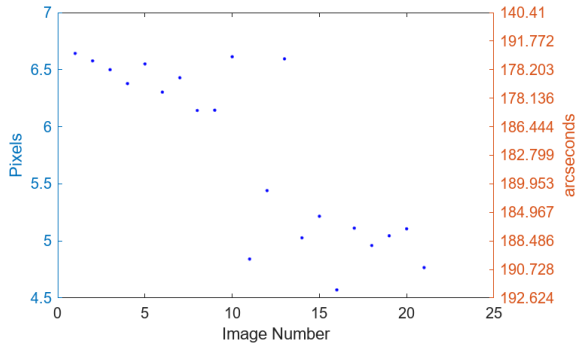

Figure 18: Distance Between Expected and Computed Moon Position.

Further assessment will be performed to account for atmospheric refraction based on the viewing conditions of these images. In future field tests, the new ODIN v1 prototype equipped with the rigid optical base should mitigate any mechanical shifting.

With these improvements in mind for future testing, the trends in the position estimation residuals are still useful to analyze. The residuals are shown in the camera frame in Fig. 19 and as range and bearing in Fig. 20. As expected with most optical measurements, range estimation, or estimation along the camera boresight, is the least accurate due to the range ambiguity when viewing a 3D object in 2D space. This behavior is well reflected in the residuals, which are much larger in range or in the Z component in the camera frame.

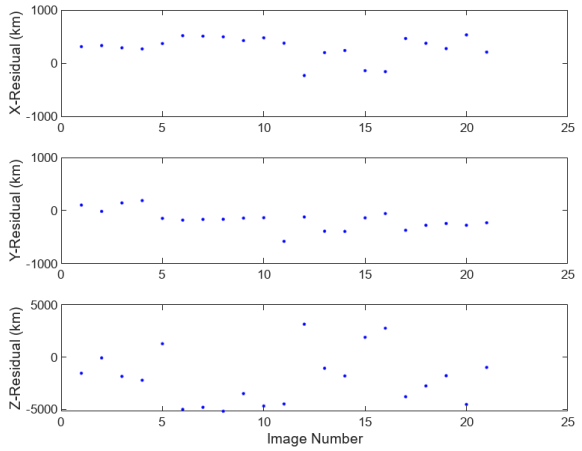

Figure 19: Position estimation residuals as shown in the camera frame. Measurements along the boresight (z-axis) have much greater residuals than the along track (x and y axes).

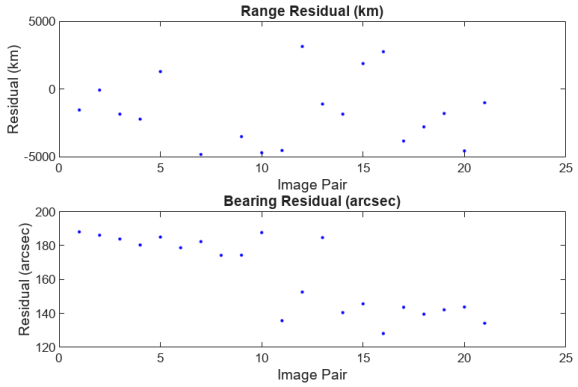

Figure 20: OpNav measurements are often much more accurate in estimation of bearing than of range to the target.

Software Application Testing

Development of the ODIN application is underway. Currently, the application can receive commands and mode into the commanded mode. The camera control software has been integrated, and ODIN can take simultaneous NAC images at the desired frequency.

FUTURE WORK

Future ODIN development and testing is planned to advance the technology to TRL 6 through a subsystem prototype demonstration in a relevant ground-based environment. This milestone will be achieved with the ODIN v1 prototype and an onboard version of the ODIN flight software at a dark sky site. The core functionalities of ODIN such as simultaneous image capture, attitude determination, and position estimation will be demonstrated in an autonomous operation mode. Other ODIN algorithms which have required more development such as camera calibration with uncalibrated star ID and OpNav target determination may be demonstrated manually. The various algorithms will be integrated into the ODIN application and full integrated software/hardware testing will complete the project.

Acknowledgments

This work was funded by the NASA University SmallSat Technology Partnership (USTP) initiative.

References

[1] Rebecca J. Inman. Demonstration of the orion optical navigation system on artemis i. *4th Space Imaging Workshop*, SIW24(10), 2024.

[2] John A. Christian. A tutorial on horizon-based optical navigation and attitude determination with space imaging systems. *IEEE Access*, 9:19819–19853, 2021.

[3] D Brown. Close-range camera calibration photo, 1971.

[4] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. 10 2006.

[5] John A. Christian. *Fundamentals of Spacecraft Optical Navigation*. John Wiley & Sons, 2025.

[6] Dustin Lang, David W. Hogg, Keir Mierle, Michael Blanton, and Sam Roweis. Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal*, 139(5):1782–1800, 2010.

[7] John A. Christian. Accurate planetary limb localization for image-based spacecraft navigation. *Journal of Spacecraft and Rockets*, 54(3):708–730, 2017.

[8] Kaasalainen, M. and Tanga, P. Photocentre offset in ultraprecise astrometry: Implications for barycentre determination and asteroid modelling. *AA*, 416(1):367–373, 2004.

[9] Charles H. Acton. Ancillary data services of nasa's navigation and ancillary information facility. *Planetary and Space Science*, 44(1):65–70, 1996. Planetary data system.

[10] Charles Acton, Nathaniel Bachman, Boris Semenov, and Edward Wright. A look towards the future in the handling of space science mission geometry. *Planetary and Space Science*, 150:9–12, 2018. Enabling Open and Interoperable Access to Planetary Science and Heliophysics Databases and Tools.