VSLAM and Vision-based Approach and Landing for Advanced Air Mobility

Evan Kawamura *[1], Chester Dolph †[2], Keerthana Kannan ‡[1], Nelson Brown §[3], Thomas Lombaerts ¶[1], and Corey Ippolito ‖[1]

[1]*NASA Ames Research Center, Moffett Field, Mountain View, CA 94035*

[2]*NASA Langley Research Center, 1 Nasa Dr, Hampton, VA 23666*

[3]*NASA Armstrong Flight Research Center, Edwards CA 93523*

Advanced Air Mobility (AAM) aircraft have many challenges in landing accurately and safely in urban, suburban, and rural environments. Localization in large and open rural environments could utilize GPS, but AAM aircraft in urban environments will encounter GPS degradation. Another challenge involves flight operation time, i.e., flying during the day or at night. There are different guidelines, landmarks, and landing light configurations at runways, heliports, and vertiports for daytime and nighttime applications. Tailoring feature detection methods for AAM approach and landing during the day and night pose different issues and challenges. It is easier to detect edges, lines, and other runway markers during the day than at night. Conversely, it is easier to see landing light configurations and patterns at nighttime than daytime. Consequently, utilizing the same feature detector for daytime and nighttime operations may not be feasible. This paper focuses on a vision-based precision approach and landing (PAL) by comparing ORB SLAM 2, a Vision Simultaneous Localization and Mapping (VSLAM) algorithm, and a novel EKF that combines onboard IMU measurements with coplanar pose from orthography and scaling with iterations (COPOSIT). Conducting unmanned aerial system (UAS) flight tests at NASA Armstrong Flight Research Center (AFRC) with landmarks and fiducials distributed around the landing zone provides a simulated AAM approach and landing data to test vision-based PAL methods to provide Alternative Position, Navigation, and Timing (APNT) solutions for AAM PAL applications. The novel vision-based PAL EKF with IMU and COPOSIT provides accurate state estimation when distributed landmarks and fiducials are in the field of view.

I. Introduction

Advanced Air Mobility (AAM) aircraft require precision approach and landing systems in several urban, suburban, and rural environments. Implementing current state-of-the-art methods approved for automated approach and landing for AAM operations is complex. This paper focuses on vision-based methods to demonstrate APNT solutions for AAM approach and landing such that having alternative APNT solutions are vital to successful AAM operations in urban environments with GPS degradation. This paper builds on previous work by conducting UAS flight tests at NASA Armstrong Flight Research Center to demonstrate vision-based APNT solutions for AAM approach and landing.

A. Feature Detection

There are many types of feature detection methods. The scale invariant feature transform (SIFT) provides invariance to image rotation and scale with robustness to affine distortion, change in 3D viewpoint, the addition of noise, and change in illumination [1]. The Speeded-Up Robust Features (SURF) feature detector resembles and generally outperforms SIFT due to SURF being less sensitive to noise [2]. The Harris corner detector finds corners and edges based on

*Computer/GNC Engineer, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

†Aerospace Engineer, Aeronautics Systems Engineering Branch, NASA Langley Research Center, 1 Nasa Dr, Hampton, VA 23666.

‡Software Engineer, KBR Wyle Services, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

§Aerospace Engineer, NASA Armstrong Flight Research Center, Edwards CA 93523.

¶Aerospace Research Engineer, KBR Wyle Services, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

‖Aerospace Scientist, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA

local auto-correlation [3]. The Features from Accelerated Segment Test (FAST) feature detector is a fast feature extraction and matching method that can operate at 400 Hz [4]. Image registration techniques align two or more images by intensities or features [5]. The Hough transform can detect circles with modifications such as edge orientation, considering a range of circle radii, using complex accumulator arrays [6, 7]. The Kaze feature detector utilizes nonlinear diffusion filtering, tends to be more computationally expensive than SIFT, but has comparable performance to SURF, SIFT, and center-surround detectors (CenSurE) in terms of detection and description [8]. A minimum eigenvalue algorithm detects corners, occlusions, disocclusions, and features not associated with points in the world in affine image transformations [9]. The Binary Robust Invariant Scalable Keypoints (BRISK) feature detector uses scaled concentric circles at keypoints, and it has similar performance to SURF but with less computational time [10]. The Oriented FAST and rotated BRIEF (ORB) feature detector operates at two orders of magnitude faster than SIFT with good performance in real-world applications such as object detecting and patch-tracking on a smartphone [11]. The maximally stable extremal regions (MSER) feature detection method has nearly linear computational time complexity and a fast detection algorithm for indoor and outdoor applications [12].

B. VSLAM Methods

There are many types of VSLAM methods because there are many feature detectors. ORB SLAM is a monocular VSLAM method that works in indoor and outdoor environments with real-time capabilities and is robust to motion clutter [13]. ORB SLAM 2 applies to monocular, stereo, and RGB-D cameras and has real-time capabilities [14]. ORB SLAM 3 applies to monocular, stereo, and RGB-D cameras with pinhole and fisheye lens models for a tightly integrated system with maximum a posteriori estimation for real-time operations in indoor and outdoor environments [15]. A visual odometry VSLAM method demonstrates accurate pose tracking with little overhead and good feature selection [16]. A distributed particle filter with an INS/VSLAM system has improved performance compared to an INS-only navigation solution [17].

Other studies directly use VSLAM methods for aircraft and robot navigation. A vision-aided inertial navigation solution for UAVs works in GPS-denied environments for closed-loop flights with Harris corner detection to extract feature points [18]. An improved monocular ORB SLAM algorithm for robots with adaptive FAST threshold and image enhancement demonstrates robust performance in complex lighting conditions [19]. Another VSLAM study compares Harris corner detection, Kanade-Lucas-Tomasi track, and SIFT in an indoor environment and concludes that the choice of feature extractor is not critical [20]. A Rao-Blackwellized particle filter for VSLAM with IMU, barometric altimeter, and monocular camera inputs leads to UAV state estimation while building a feature map [21]. Another study evaluates and tests several monocular VSLAM methods for fixed-wing aircraft: MonoSLAM, PTAM, OKVIS, LSDSLAM, ORB SLAM 2, and SVO [22]. Binocular VSLAM for quadrotor UAVs demonstrates improved performance compared to the traditional VSLAM framework [23].

C. Overview of Work

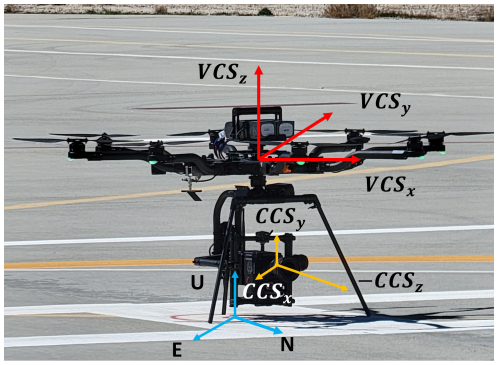
UAS flight tests at the NASA Armstrong Flight Research Center (AFRC) helipad provide INS/GPS trajectory logs to serve as ground truth for both APNT solutions. Preliminary testing with several feature detection and VSLAM methods on the experimental data showed that ORB SLAM 2 and Harris corner detection work well for the obtained dataset. This paper compares two APNT solutions: ORB SLAM 2 and VAL (COPOSIT-EKF), an EKF that combines coplanar POSIT and IMU measurements from Ref. [24]. Cones and the helipad markings distributed around the landing zone serve as fiducials for visual aids and reference points for COPOSIT-EKF since it requires known feature points. Conversely, ORB SLAM 2 does not require any known a priori features but includes the fiducials to have an equivalent dataset to compare against it.

The organization of the rest of the paper is as follows. Section II has details of the Alta8 UAS flight tests at AFRC and the important landmarks and fiducials. Section III discusses details about feature detection for Vision-based Approach and Landing (VAL), and section IV shows preliminary and initial tests of VSLAM methods. Section V summarizes the EKF design, which resembles the architecture in Ref. [24, 25], and section VI summarizes the results. Finally, section VII ends with concluding remarks.

II. AFRC Flight Tests and Landmarks

UAV Alta8 flight tests at NASA Armstrong Flight Research Center provide experimental data for mock AAM approach and landing operations. Figure 1a shows the AFRC Alta8 octocopter with the inertial world coordinates (ENU), vehicle coordinates (VCS), and camera coordinates (CCS). Figure 1b shows that the center of mass of the Alta8 is approximately 17 inches (0.43 m) above the center of the camera. Thus, the estimated position vector from the Alta8 body frame to the camera frame is:

$$\boldsymbol{p}_{vcs2ccs} = \left[0, 0, -0.43\right]^T$$



(a) AFRC Alta8 Octocopter Frames

(b) AFRC Alta8 Octocopter with ruler markings in inches

Fig. 1 NASA AFRC Alta8

The state vector for the Alta8 UAS is:

$$\mathbf{s} = \begin{bmatrix} E & N & U & v_N & v_E & v_U & \phi & \theta & \psi \end{bmatrix} \tag{1}$$

such that E, N, U are the East, North, Up coordinates in the world coordinates (inertial frame), v_E, v_N, v_U are the East, North, Up velocities in the world coordinates (inertial frame), and ϕ, θ, ψ are the roll, pitch, and yaw Euler angles.

The National Geospatial-Intelligence Agency (NGA) provided the WGS84 coordinates (latitude, longitude, and ellipsoid height) of the AFRC helipad markings with an accuracy of 0.02 m horizontally. The vertical accuracy is about 0.1 m by taking the standard deviation of the WGS84 ellipsoid heights for the surveyed points. Their geodetic surveys include stations, helipads, and vertipads [26]. The Alta8 UAS flight tests were conducted at the helipad with incoming trajectories over the lakebed. The cone locations coincide with the concrete square intersection points to yield precise, repeatable locations (see Fig. 2).

Fig. 2 NASA AFRC Helipad Cones: locations coincide with the intersection points for precise locations

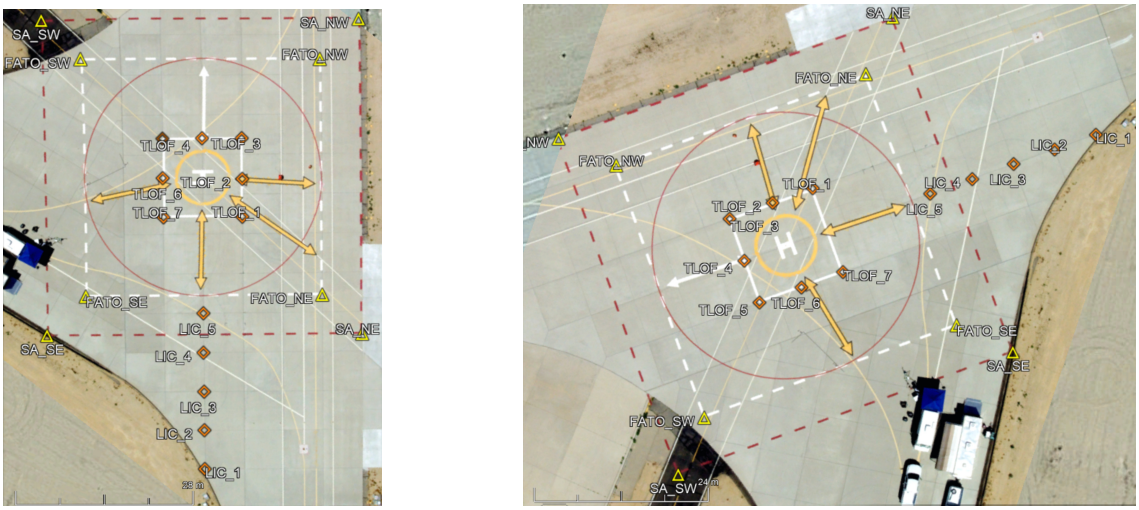
It is possible to estimate the Pulse Light Approach Slope Indicator (PLASI) location based on the known NGA-measured sites of the AFRC helipad markings. However, the PLASI was not included as a critical landmark since its location varied between flight tests.

Table 1 AFRC Helipad Dimensions

	side (ft)	side (m)	dashes (ft)	dashes (m)	gaps (ft)	gaps (m)
TLOF	40	12.2	N/A	N/A	N/A	N/A
FATO	120	36.6	5	1.5	5	1.5
SA	160	48.8	5	1.5	6	1.8

Figure 3 shows Google Earth images of the helipad at NASA AFRC, its markings, and fiducial (cone) locations. Table 1 summarizes the AFRC helipad dimensions. The corner dashes of the FATO are slightly longer at 7.5 ft (2.3 m).

Predicting the landmark and fiducial locations in pixel coordinates begins by importing the latitude and longitude coordinates into MATLAB. Then, converting them into model coordinates (MCS) with the helipad as the origin with MCS aligning with the Alta8 path during approach and landing. Then, rotating MCS into world coordinates (WCS) aligns the landmarks and fiducials with true north (see Fig. 4). Comparing Fig. 3 with Fig. 4 provides a visual confirmation that the coordinates for the landmarks and fiducial are correct. The camera model uses the landmarks and fiducials' MCS and WCS coordinates to determine their pixel coordinates. Table 2 shows the landmark and fiducial coordinates with East and North (WCS) relative to the center of the helipad.



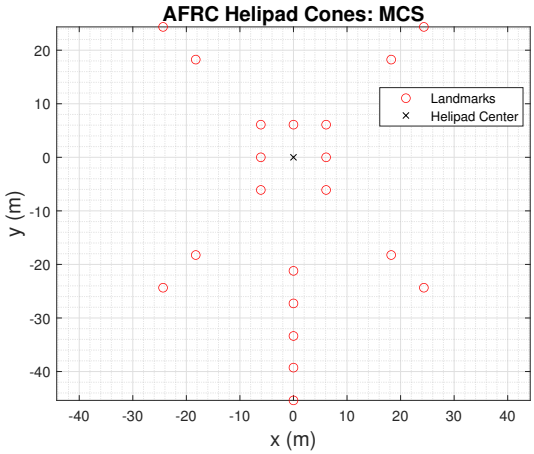
(a) AFRC Helipad Aligned with Atla8 Trajectory **(b) AFRC Helipad Aligned with True North**

Fig. 3 NASA AFRC Helipad Markings and Fiducials Alignment: Alta8 Trajectory & True North, cardinal directions with respect to true north

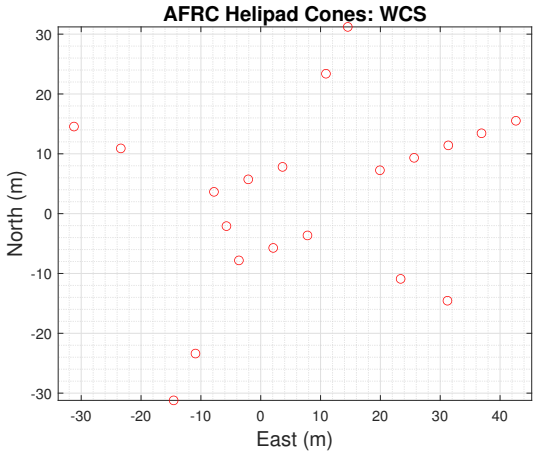
Figure 5 shows the Alta8 trajectory in Google Earth. The Alta8 initially starts over the lakebed, facing the helipad. The pilot hovers for a few seconds and then descends toward the helipad to resemble an AAM aircraft glideslope approach and landing profile. The Alta8 approach and landing profile does not perfectly match the prescribed 9° AAM glideslope presented in Ref. [27] because the Alta8 Pixhawk flight controller lacks a glideslope controller.

III. Feature Detection for VAL

This section describes the feature detection methods to identify the AFRC helipad markings and fiducials. Refs. [24, 25] utilize Hough circle detection to find the landing lights, but it might not be an ideal feature detection method for the AFRC helipad due to the sharp corners and cones. Figure 6 shows an initial comparison between Hough circle and Harris corner detection of one of the frames during the Alta8 approach and landing. Fig. 6a shows that Hough circle detection does not detect all twelve of the orange traffic cones, while Harris corner detection does detect all twelve of the orange traffic cones, including the PLASI, as shown in Fig. 6b. Thus, Hough circle detection might not be an ideal feature detector to implement for AFRC helipad feature detection. On the other hand, Harris corner detection identifies the key landmarks in the scenery. It detects all twelve orange traffic cones, making it a more suitable feature detector.



(a) AFRC Helipad Markings with Cones (MCS)



(b) AFRC Helipad Markings with Cones (MATLAB)

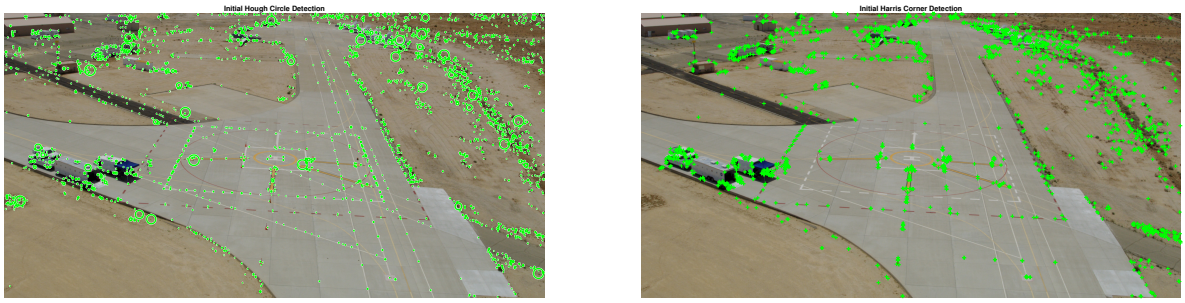
Fig. 4 NASA AFRC Helipad Markings and Fiducials in MCS & WCS (MATLAB)

Table 2 AFRC Landmark and Fiducial Coordinates

Fiducial Name	Abbreviation	Latitude (°)	Longitude (°)	East (m)	North (m)
TLOF Cone 1	$TLOF_1$	34.959163	-117.881696	3.64	7.81
TLOF Cone 2	$TLOF_2$	34.959145	-117.881757	-2.09	5.73
TLOF Cone 3	$TLOF_3$	34.959126	-117.881822	-7.81	3.64
TLOF Cone 4	$TLOF_4$	34.959074	-117.881799	-5.73	-2.09
TLOF Cone 5	$TLOF_5$	34.959022	-117.881776	-3.64	-7.81
TLOF Cone 6	$TLOF_6$	34.959041	-117.881713	2.09	-5.73
TLOF Cone 7	$TLOF_7$	34.959060	-117.881651	7.81	-3.64
Lead In Cone 1	LIC_1	34.959230	-117.881270	42.7	15.5
Lead In Cone 2	LIC_2	34.959211	-117.881333	36.9	13.4
Lead In Cone 3	LIC_3	34.959193	-117.881394	31.4	11.4
Lead In Cone 4	LIC_4	34.959175	-117.881457	25.6	9.33
Lead In Cone 5	LIC_5	34.959157	-117.881520	19.9	7.25
FATO Northeast Corner	$FATO_{NE}$	34.959304	-117.881617	10.9	23.4
FATO Northwest Corner	$FATO_{NW}$	34.959192	-117.881993	-23.4	10.9
FATO Southwest Corner	$FATO_{SW}$	34.958882	-117.881857	-10.9	-23.4
FATO Southeast Corner	$FATO_{SE}$	34.958994	-117.881480	23.4	-10.9
Safety Area Northeast Corner	SA_{NE}	34.959374	-117.881576	14.5	31.2
Safety Area Northwest Corner	SA_{NW}	34.959225	-117.882078	-31.2	14.6
Safety Area Southwest Corner	SA_{SW}	34.958812	-117.881897	-14.6	-31.2
Safety Area Southeast Corner	SA_{SE}	34.958961	-117.881396	31.2	-14.6

Fig. 5 Alta8 Flight Test Trajectory: Google Earth

Therefore, this paper uses Harris corner detection instead of Hough circle detection, while the COPOSIT-EKFs in Refs. [24, 25] use Hough circle detection.



(a) AFRC Flight Test: Initial Hough Circle Detection **(b) AFRC Flight Test: Initial Harris Corner Detection**

Fig. 6 Initial Comparison: Harris Corner and Hough Circle Detection

IV. VSLAM Method Selection

Every VSLAM algorithm follows a sequence of map initialization, tracking, local mapping, and loop closing (see Fig. 7). There are many VSLAM algorithms, but this paper focuses on applying ORB SLAM and ORB SLAM 2 for the AFRC helipad Alta8 approach and landing.

Figure 8 compares initial results between ORB SLAM and ORB SLAM 2 at different frames. ORB SLAM tends to find numerous features not associated with the helipad markings: the truck, buildings, fence, fighter jets, and terrain. Note that the default bag of features in ORB SLAM lacked helipad markings since it was initially tailored for features and objects inside an office *.

Conversely, ORB SLAM 2 tends to find numerous features associated with the helipad markings. ORB SLAM implements a bag of words/features approach, which was not tuned to helipad markings [13]. Contrarily, ORB SLAM 2 is more feature based than ORB SLAM and picks up more features than ORB SLAM, which provides better performance [14].

*https://tinyurl.com/orbslam

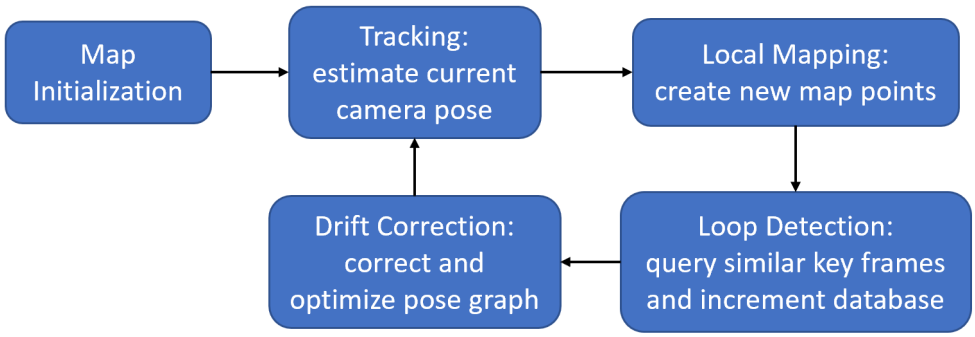
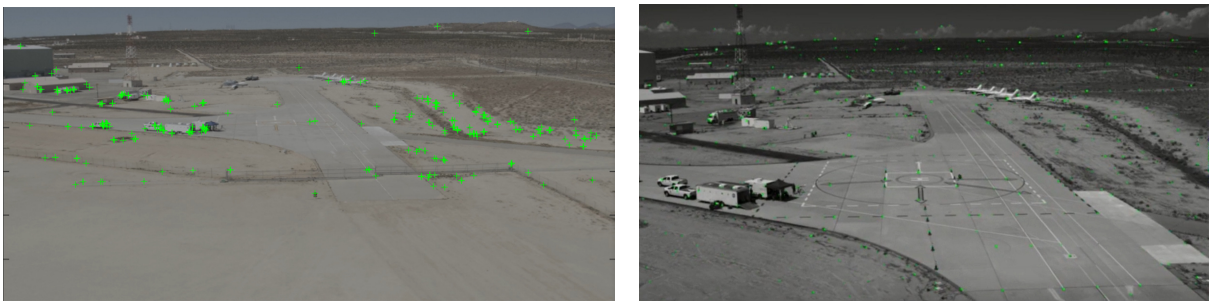


Fig. 7 General VSLAM Block Diagram



(a) AFRC Flight Test: ORB SLAM **(b) AFRC Flight Test: ORB SLAM 2**

Fig. 8 ORB SLAM Initial Comparison

Table 3 shows the main ORB SLAM and ORB SLAM 2 parameters. NL is the number of decomposition levels, and NP is the number of points such that higher resolutions require more points for feature extraction. The outlier thresholds for computing the homography and fundamental matrices are OT_h and OT_{FM}, respectively. Finally, the ORB SLAM 2 initial and minimum extractor thresholds are OT_{init} and OT_{min}, respectively. Since ORB SLAM uses a bag of features not tailored to helipad markings, tuning these parameters did not yield better performance.

Table 3 ORB SLAM Parameters

	Scale Factor	NL	NP	OT_h	OT_{FM}	OT_{init}	OT_{min}
ORB SLAM	1.2	8	1000	10	4	n/a	n/a
ORB SLAM 2	1.2	8	2000	n/a	n/a	12	4

Preliminary ORB SLAM tests had scaling ambiguity, so they could not optimize its estimated trajectory. VSLAM lacks scale without resolving scale ambiguity, so the estimated trajectory remains unscaled if scale ambiguity is not solved. Fig. 9 shows the ORB SLAM estimated trajectory. ORB SLAM also neglects numerous relevant features, such as the helipad markings, due to limited objects in its bag of features.

Contrarily, Fig. 10 shows a reasonable estimate of the unscaled ORB SLAM 2 trajectory because it tracks more features associated with the helipad and its markings. Testing this ORB SLAM example with its default bag of features also on AFRC april tags for camera calibration showed similar results with ORB SLAM preferences towards tracking the buildings, automobiles, and other features that were not associated with the april tags. Overall, adding helipad markings into ORB SLAM's bag of features would yield better performance. However, ORB SLAM 2 includes more features and performs better in these preliminary tests, so ORB SLAM 2 is the selected VSLAM method to compare against the Alta8 approach and landing trajectory.

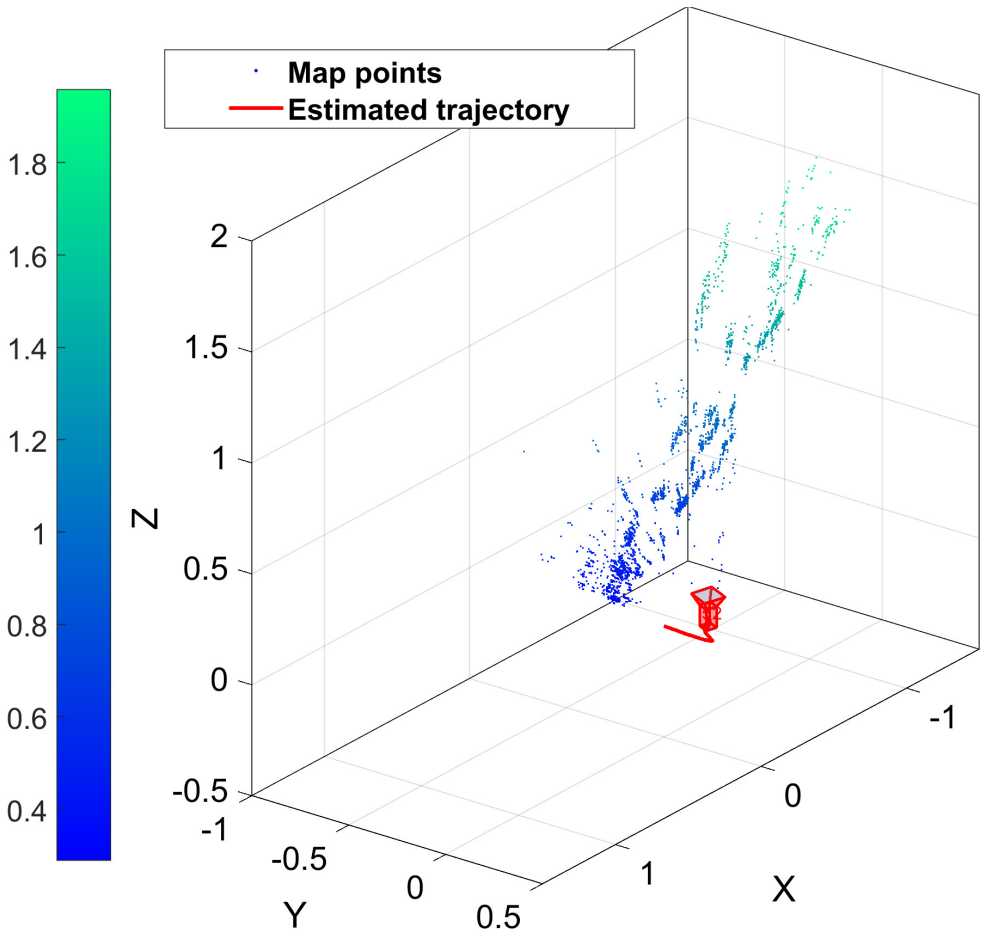


Fig. 9 AFRC Flight Test: Preliminary ORB SLAM Unscaled Map Points and Estimated Trajectory (generated in MATLAB)

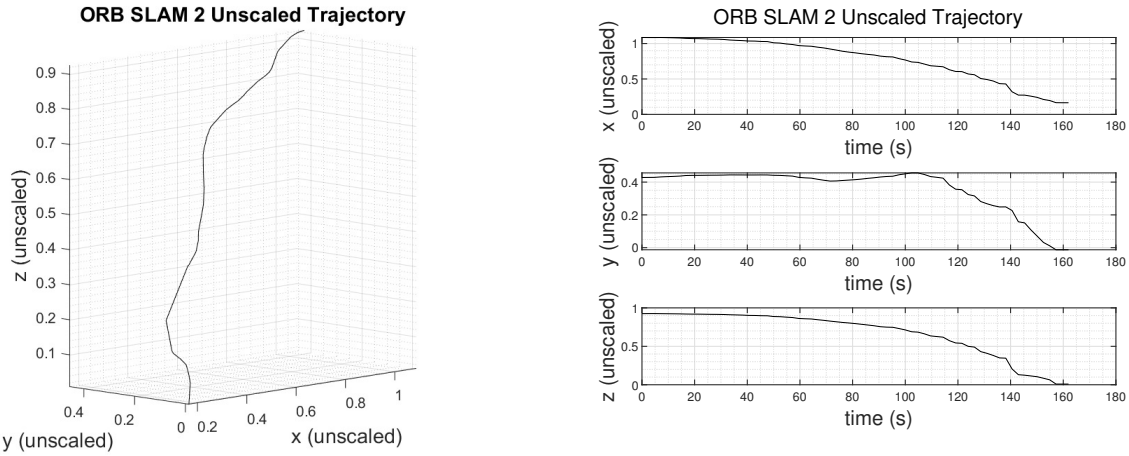


Fig. 10 AFRC Flight Test: ORB SLAM 2 unscaled and estimated trajectory without map points (generated in Python, plotted in MATLAB)

V. VAL Design

The VAL design for this paper follows the design and details provided in Ref. [25] with some differences:

1) The timestep for this paper is 2.38 seconds to match the ORB SLAM 2 time step, which gives a more accurate comparison. Taking multiple measurements in a short amount of time increases the certainty of the state estimation, which leads to higher accuracy.
2) The **Q** and **R** matrix components contain different values (see VI). The process noise covariance includes uses the IMU measurement variances:

$$\mathbf{Q} = diag\left[\sigma_{acc,x}^2 \quad \sigma_{acc,y}^2 \quad \sigma_{acc,z}^2 \quad \sigma_{gyro,x}^2 \quad \sigma_{gyro,y}^2 \quad \sigma_{gyro,z}^2\right]. \tag{2}$$

The measurement noise covariance utilizes the variances from the coplanar POSIT algorithm:

$$\mathbf{R} = diag\left[\sigma_E^2 \quad \sigma_N^2 \quad \sigma_U^2 \quad \sigma_{vE}^2 \quad \sigma_{vN}^2 \quad \sigma_{vU}^2 \quad \sigma_\phi^2 \quad \sigma_\theta^2 \quad \sigma_\psi^2\right]. \tag{3}$$

The COPOSIT velocity measurements depend on the COPOSIT position measurements, so they are not completely independent [25]:

$$\boldsymbol{v}_{coposit,i} = \frac{\boldsymbol{p}_{coposit,i} - \boldsymbol{p}_{coposit,i-1}}{t_i - t_{i-1}}. \tag{4}$$

3) This paper uses Harris corner detection instead of Hough circle detection. Figure 11 shows the VAL block diagram with Harris corner detection instead of Hough circle detection (see the red text above the Estimated Detected Landing Lights (PC): Image Processing block).

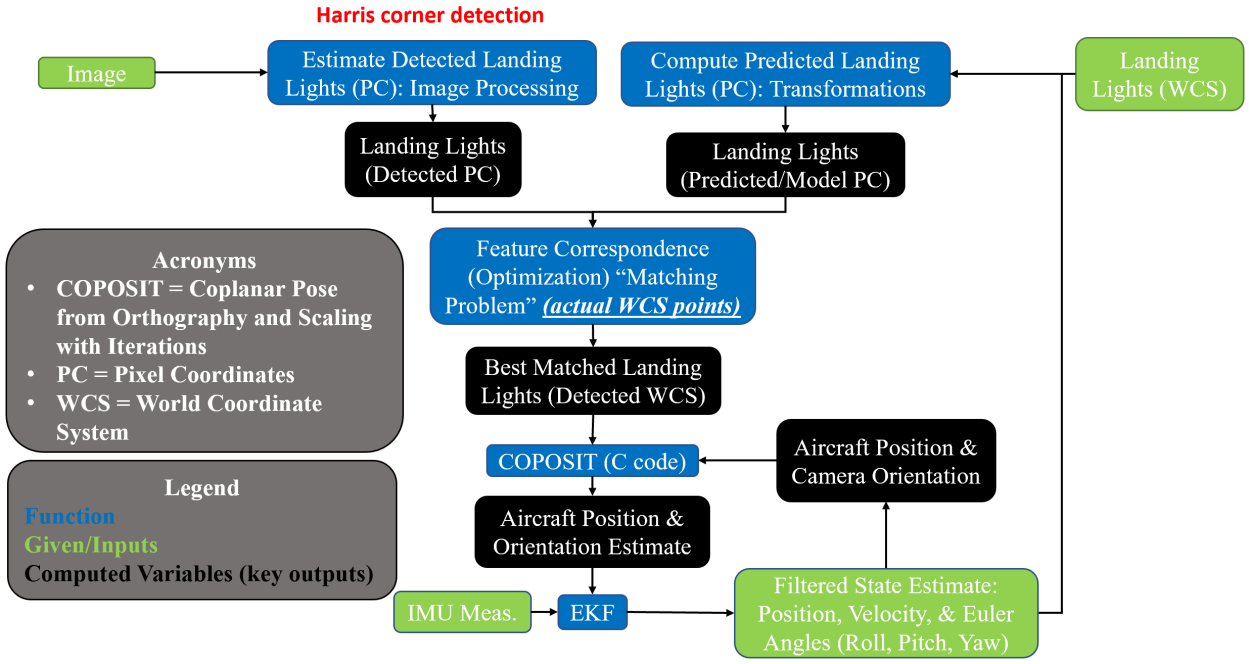


Fig. 11 VAL Block Diagram with Harris Corner Detection, modified from Ref. [25]

Figure 12 shows the rates, inputs, and outputs of the main components high-level block diagram of the Vision-based Approach and Landing System (VALS) from Ref. [25]. This paper uses the EKF presented in Ref. [25], which shows a high-level block diagram of the EKF structure (see Fig. 13).

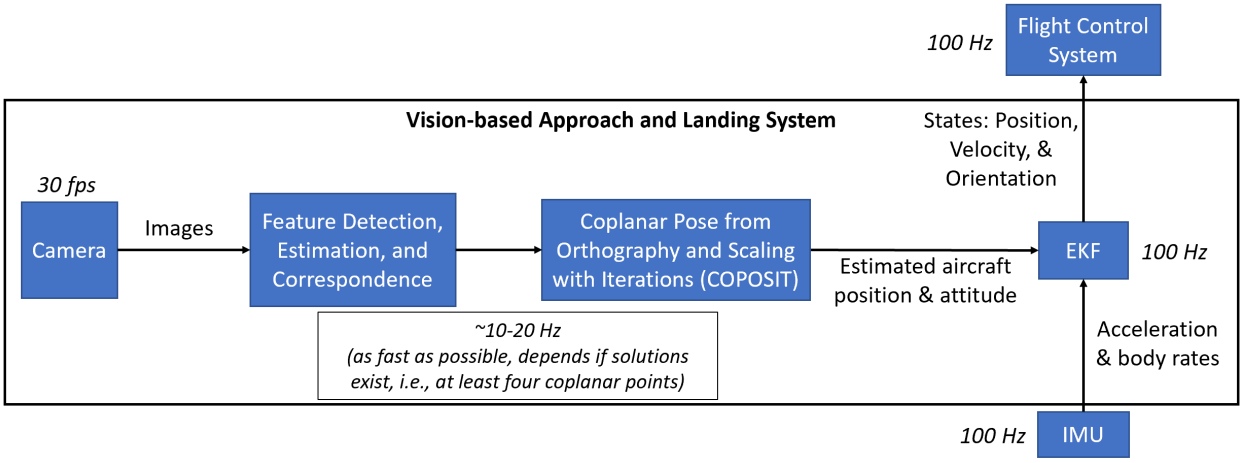


Fig. 12 Vision-based Approach and Landing System (VALS) Block Diagram [25]

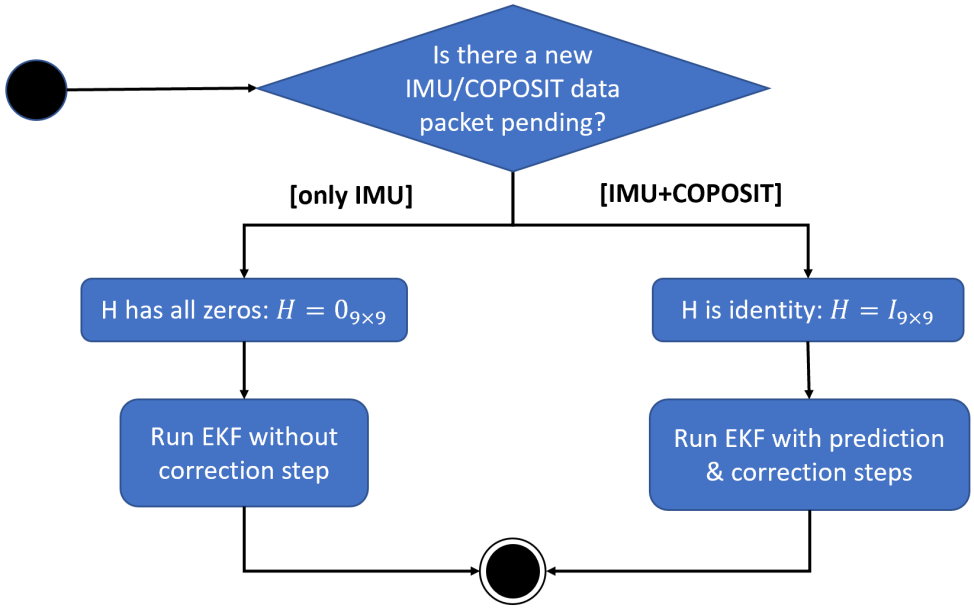


Fig. 13 EKF Flowchart: 1) running the EKF without a correction step and only the prediction step (dead reckoning) occurs since the Kalman gain, K_k, and measurement innovation, v_k, equal zero): $\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-)$; 2) running the EKF with both prediction and correction steps has nonzero values for the Kalman gain, K_k, and measurement innovation, v_k: $\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-) + K_k v_k$ [25]

VI. Results

This section shows the images' time synchronization, the telemetry data's accuracy, camera parameters, and the results for the APNT solutions: ORB SLAM 2 and COPOSIT-EKF.

A. Time Synchronization

The RED camera frame rate is 60 fps, which means one frame occurs every $0.01\overline{6}$ seconds. One of the Alta8 preflight checklist items includes placing a clock in the camera's view to match the frame and times post-flight based on the camera frame rate of 60 fps. Choosing frame 25939 at the top of the glideslope occurs at 10:54:28 AM (local time, Pacific Daylight Time (UTC-7)) based on the physical time hack with the clock displayed during the preflight steps. Extracting 9720 frames (25939 to 35659) from the camera through the REDLine command line utility provided by the camera manufacturer [†], which gives 162 seconds (2.7 min) of footage. The Alta8 completes a mock AAM approach and landing trajectory during this time, as shown earlier in Fig. 5. Thus, the trimmed Pixhawk ulog time histories and ORB SLAM 2 data sets run for 162 seconds to match the relevant segment of extracted frames.

B. RED Camera Parameters

Table 4 shows the RED camera parameters. The camera pitch down angle is approximately $-5°$ throughout the flight. Ref. [25] includes a far plane in its camera parameters since it uses the perspective matrix and frustum (matrix in the glFrustum function [‡]) in its camera model for predicting landmarks. This paper also includes the far plane and uses the same value even though the far plane extends much farther than 4000 m in the real world.

Table 4 RED Camera Parameters

Parameter	Image Size	Pitch Angle (°)	Near Plane (m)	Far Plane (m)	FOV_h (°)	FOV_v (°)
Value	4096×2160	-5	1	4000	84	44.30

C. GPS Accuracy

Table 5 shows the GPS accuracy of the Alta8 Pixhawk telemetry data in terms of horizontal dilution of precision (HDOP), vertical dilution of precision (VDOP), and the number of satellites. Overall, the Alta8 flight test had excellent satellite coverage, ranging from 17 to 20. The low HDOP and VDOP values demonstrate accurate horizontal and vertical GPS accuracy. The Alta8 has a ublox M8N GPS receiver and Here2 antenna, so its accuracy notionally ranges from 2 to 4 m horizontally [§].

Table 5 GPS Accuracy: HDOP, PDOP, & Number of Satellites

	HDOP	VDOP	Number of Satellites
Median	0.590	0.810	20
Average	0.596	0.815	19.63
Min	0.590	0.800	17
Max	0.650	0.930	20
Std	0.0106	0.0146	0.566

D. ORB SLAM 2

ORB SLAM 2 yields the unscaled position of the generated trajectory. Syncing the Pixhawk data with the ORB SLAM 2 data occurs before scaling (recall subsection VI.A). The scaled data follows the linear y-intercept form:

$$\boldsymbol{p}_{scaled} = A\boldsymbol{p}_{unscaled} + b \tag{5}$$

[†] https://www.red.com/download/redline-linux-beta

[‡] https://learn.microsoft.com/en-us/windows/win32/opengl/glfrustum

[§] https://content.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_UBX-15031086.pdf

such that A, b are the slope and y-intercept values, respectively. Applying LP Simplex in the Excel solver solves for the slope and y-intercept of the scaled ORB SLAM 2 position values while minimizing the sum of the squared error. The squared difference between the Pixhawk and scaled ORB SLAM 2 data with the time step ($\Delta t = t_{i+1} - t_i$) is:

$$\begin{aligned} \Delta x^2 \Delta t &= (x_{scaled,wcs} - x_{pixhawk,wcs})^2 \Delta t, \\ \Delta y^2 \Delta t &= (y_{scaled,wcs} - y_{pixhawk,wcs})^2 \Delta t, \\ \Delta z^2 \Delta t &= (z_{scaled,wcs} - z_{pixhawk,wcs})^2 \Delta t. \end{aligned} \tag{6}$$

The sum of the squared difference multiplied by the time step is:

$$\Gamma = \sum_{i=0}^{N-2} (\boldsymbol{p}_{scaled,i} - \boldsymbol{p}_{pixhawk,i})^2 \Delta t. \tag{7}$$

in which N is the number of points. Breaking down Eqn. (7) into its x,y,z components yields:

$$\begin{aligned} \epsilon_x &= \sum_{i=0}^{N-2} (x_{scaled,i} - x_{pixhawk,i})^2 \Delta t, \\ \epsilon_y &= \sum_{i=0}^{N-2} (y_{scaled,i} - y_{pixhawk,i})^2 \Delta t, \\ \epsilon_z &= \sum_{i=0}^{N-2} (z_{scaled,i} - z_{pixhawk,i})^2 \Delta t. \end{aligned} \tag{8}$$

The rounded values for $\epsilon_x, \epsilon_y, \epsilon_z$ from Eqn. (8) are 173827, 1438996, and 18456, respectively.

Figure 14a compares the x component of the Pixhawk and ORB SLAM trajectories. The x values from the ORB SLAM 2 trajectory are inaccurate and have some drift. However, the error based on Eqn. (6) decreases towards the end of the trajectory, as shown in Fig. 14b. The rounded value for ϵ_x from Eqn. (8) is 173,827.

Similarly, Fig. 15a compares the y component of the Pixhawk and ORB SLAM trajectories. The y values from the ORB SLAM 2 trajectory also tend to be inaccurate but converge with the Pixhawk y values. However, the error based on Eqn. (6) decreases towards the end of the trajectory, as shown in Fig. 15b. The rounded value for ϵ_y from Eqn. (8) is 1,438,996, which is significantly larger than ϵ_x despite converging towards the same values at the end of the trajectory.

Finally, Fig. 16a compares the z component of the Pixhawk and ORB SLAM trajectories. The z values from the ORB SLAM 2 trajectory are the most accurate and converge with the Pixhawk z values. However, the error based on Eqn. (6) decreases towards the end of the trajectory, as shown in Fig. 16b. The rounded value for ϵ_z from Eqn. (8) is 18,456, which is significantly smaller than ϵ_x and ϵ_y.

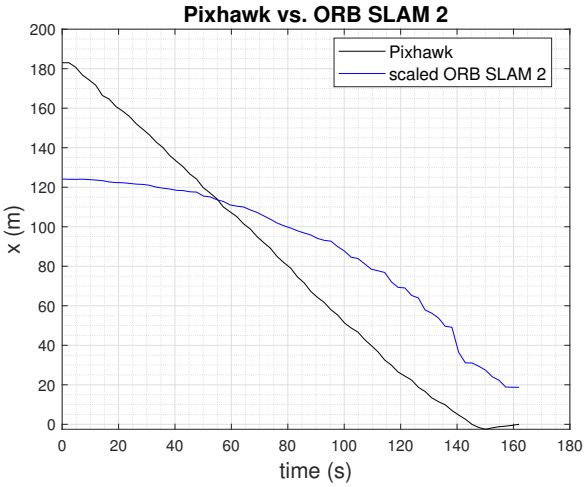
Figure 17 shows the absolute difference between the position vectors for ORB SLAM 2 and the Pixhawk telemetry data. It follows the trends of figures 14b, 15b, and 16b. The error decreases, increases, and decreases as the UAS reaches the end of the approach and landing trajectory. It is uncertain what causes the error to increase around 67 seconds, especially since the helipad and fiducials are clearly within the field of view. Future work can include investigating the feature detection methods for VSLAM. Studying alternative feature detection methods such as Harris corner detection may lead to developing VSLAM with Harris corner detection, which Ref. [18] utilizes for their vision-aided inertial navigation.

E. VAL (COPOSIT-EKF)

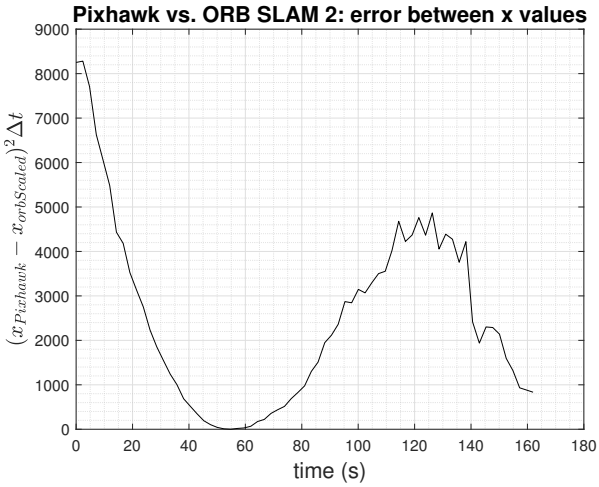
1. Feature Detection and Correspondence

Figures 18-20 show preliminary tests for feature detection and correspondence. Figure 18a shows the first frame during the AFRC helipad approach and landing with the predicted landmark locations, which is the blue block, Compute Predicted Landing Lights (PC): Transformations, in Fig. 11. Figure 18b shows the grayscale version of the first frame, which occurs before applying Harris corner detection and is the first step in the blue block, Estimate Detected Landing Lights (PC): Image Processing, of Fig. 11.

Figure 19a shows the results of applying Hough circle detection to the grayscale image and determining the closest matches to the predicted landmarks. Next, Fig. 19b shows the results of the bipartite graph matching between the

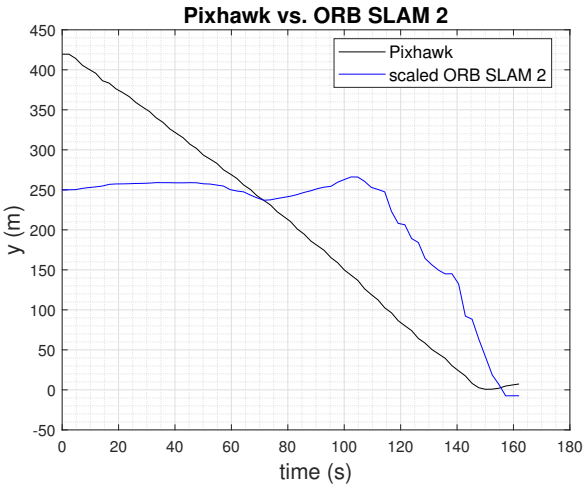


(a) Comparison: Pixhawk vs. ORB SLAM 2 x values

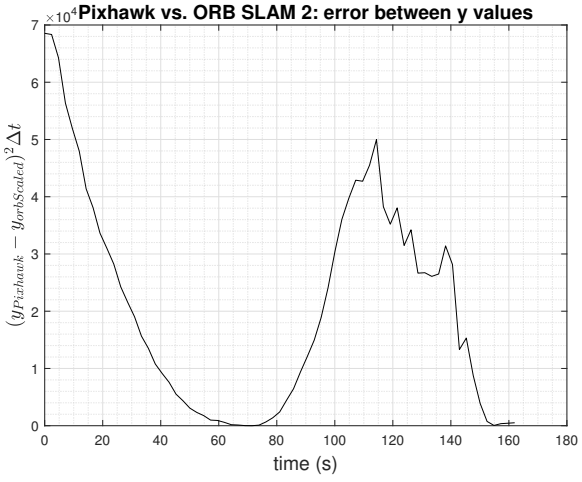


(b) Pixhawk vs. ORB SLAM 2: squared error for x

Fig. 14 ORB SLAM 2 vs. Pixhawk: x values

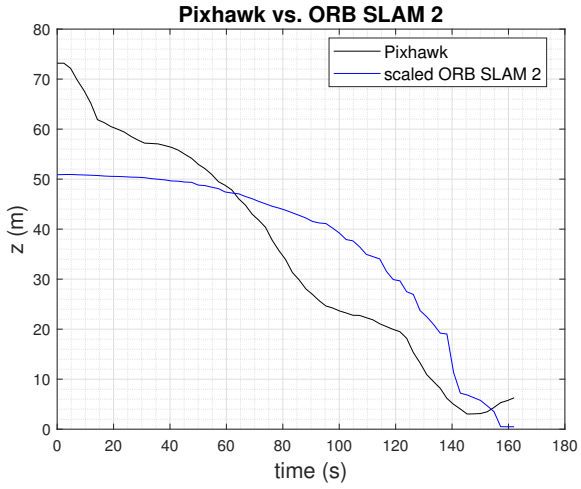


(a) Comparison: Pixhawk vs. ORB SLAM 2 y values

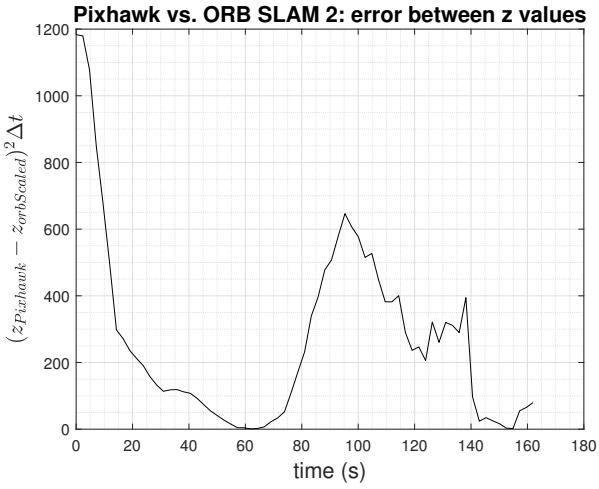


(b) Pixhawk vs. ORB SLAM 2: squared error for y

Fig. 15 ORB SLAM 2 vs. Pixhawk: y values



(a) Comparison: Pixhawk vs. ORB SLAM 2 z values



(b) Pixhawk vs. ORB SLAM 2: squared error for z

Fig. 16 ORB SLAM 2 vs. Pixhawk: z

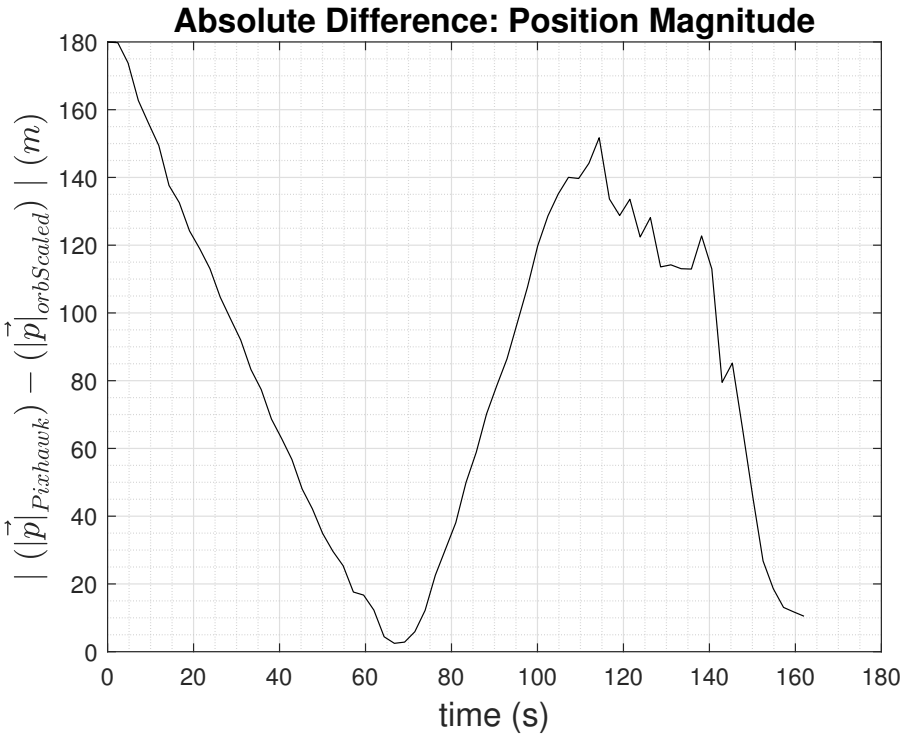
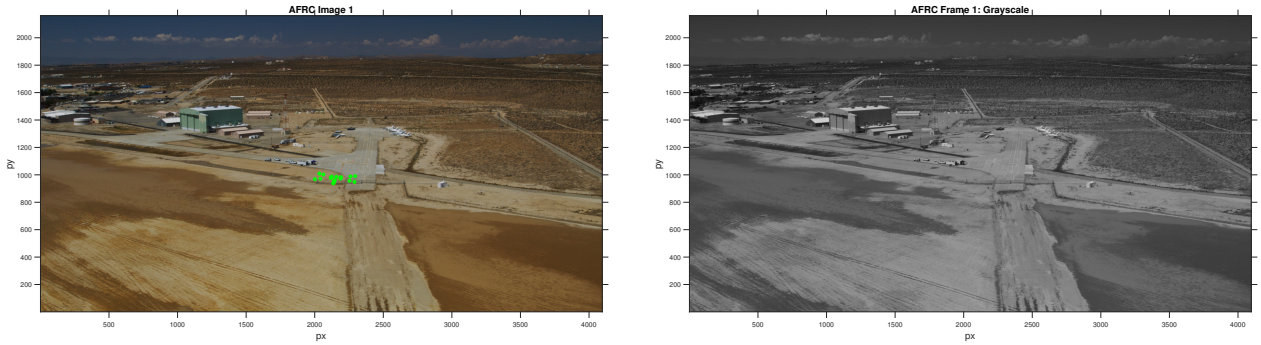


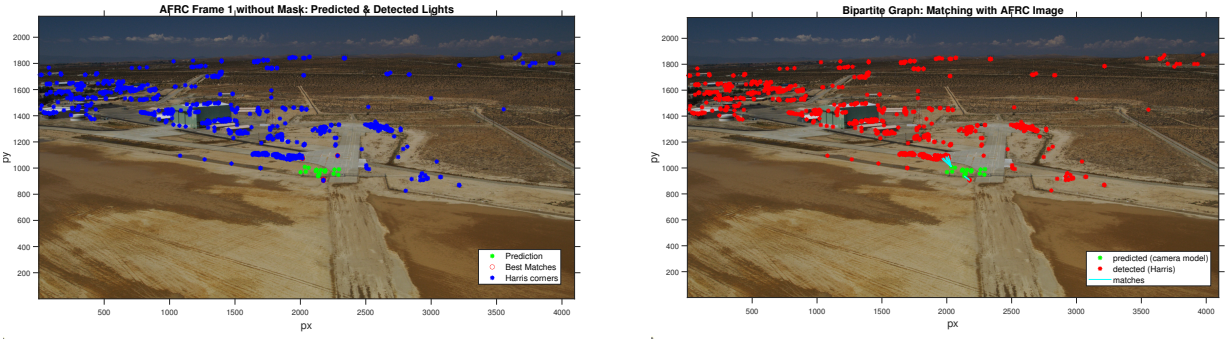
Fig. 17 Absolute Difference Between Position Magnitude of ORB SLAM 2 and Pixhawk



(a) Predicted Landmarks **(b) Grayscale before feature detection**

Fig. 18 AFRC Helipad: Predicted Landmarks & Grayscale

predicted and estimated landmarks. The blue lines connect the closest matches between the predicted and estimated landmarks based on a radial pixel tuning parameter, Δp_r. This parameter creates a circle of radius, Δp_r, around each estimated landmark (green), and the detected pixels (red) within this circle are the closest matches. Consequently, further pruning of the matches occurs if the detections fall outside this radius, i.e., Fig. 19a has three closest matches. In comparison, Fig. 19b has only two matches.

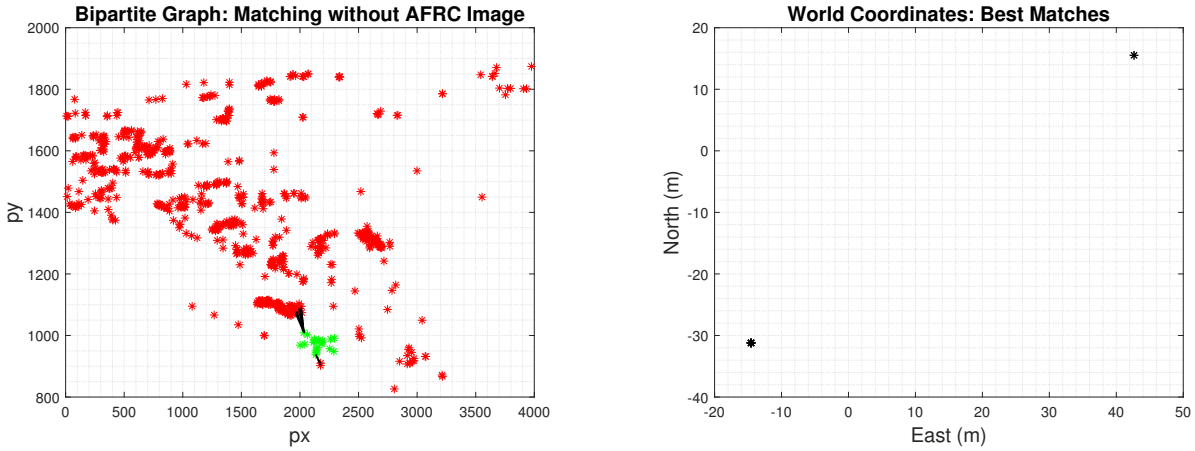


(a) Predicted and Detected Landmarks **(b) Bipartite graph for matching**

Fig. 19 AFRC Helipad with Predicted and Detected Landmarks: a) best matches before applying the radial pixel tuning parameter; b) best matches after applying the radial pixel tuning parameter

Figure 20a shows the same results as Fig. 19b but without the image in the background to see the matches between the predicted and detected landmarks. Figure 20b shows the best matches in WCS. These two matches are the bottom lead in cone (LIC_1) and southwest safety area corner (SA_{SW}), which match the Google Earth landmark locations in Fig. 3b and correspond with their values in Table 2.

Figure 21 shows the final results of the number of detections vs. time throughout the Alta8 approach and landing after tuning. COPOSIT requires at least four coplanar points (four detections) to yield a solution [28]. If there are less than four coplanar points (four detections), the EKF in VAL lacks a COPOSIT measurement. Consequently, dead reckoning occurs with only a prediction step in the EKF. Therefore, the correction and prediction steps in the EKF are identical due to the Kalman gain and measurement innovation being zero from the lack of COPOSIT measurement, i.e., $\mathbf{H} = 0_{9\times 9}$. Overall, there will be drift and inaccurate state estimation during the instances with less than four detected points and no COPOSIT measurements. However, the subsequent COPOSIT measurement will correct the divergence to create sawtooth patterns in the state estimation and covariance [25].



(a) Bipartite graph without background image **(b) AFRC Helipad Best Matches (WCS)**

Fig. 20 AFRC Helipad Best Matches: Bipartite Graph & WCS

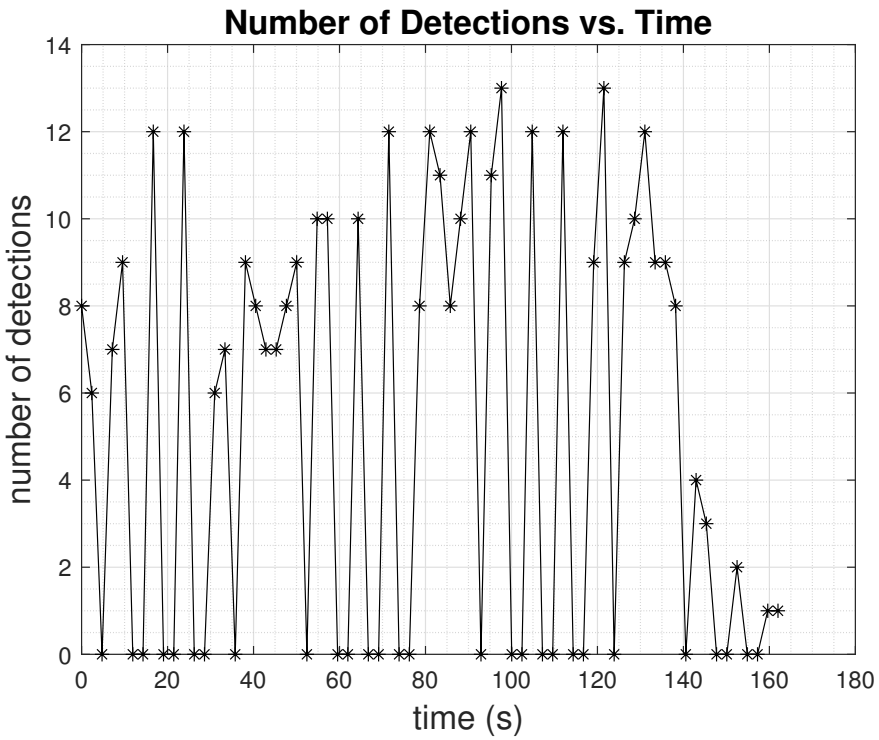


Fig. 21 Number of Detections vs. Time: COPOSIT requires at least four coplanar points (detections), so the instances of time with less than four detections cause the state estimation to drift.

2. EKF Overview

The process noise covariance (**Q**) and measurement noise covariance (**R**) values are:

$$\begin{aligned}\mathbf{Q} &= diag\begin{bmatrix}25 & 25 & 25 & 0.25 & 0.25 & 0.25\end{bmatrix}\\ \mathbf{R} &= diag\begin{bmatrix}25 & 25 & 25 & 0.09 & 0.09 & 0.09 & 0.0001 & 0.0001 & 0.0001\end{bmatrix}.\end{aligned} \tag{9}$$

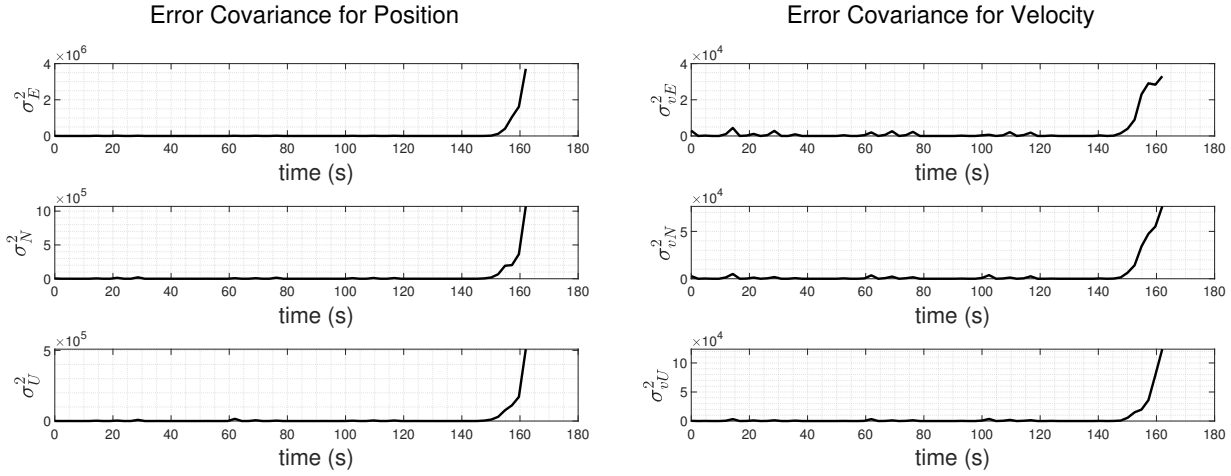
Table 6 shows the statistics of the runtime for the main VAL functions: COPOSIT, feature detection & correspondence, and EKF. Similar trends follow Ref. [25] in that the feature detection and correspondence functions take the most time, EKF runs the fastest on the order of milliseconds, and the COPOSIT function takes about hundredths of a second.

Table 6 VAL Runtime

Process	Mean (s)	Median (s)	Min (s)	Max (s)	Std (s)
COPOSIT	0.0515	0.0492	0.0463	0.0835	0.00708
Feature Detection & Correspondence	0.796	0.794	0.730	0.894	0.0399
EKF	0.00499	0.00105	0.000348	0.0741	0.0135

3. Error Covariance

Figure 22 shows the error covariance for the position and velocity of the UAS approach and landing. Divergence occurs towards the end due to the lack of landmarks and fiducials within the field of view. Consequently, COPOSIT measurements do not exist, so VAL applies dead reckoning. Therefore, the uncertainty in the estimation increases, i.e., the confidence in the estimation decreases.



(a) Error Covariance: Position **(b) Error Covariance: Velocity**

Fig. 22 Error Covariance: Position & Velocity

Similarly, Fig. 23 shows the error covariance for the Euler angles. The lack of COPOSIT measurements produces divergence (20) but is not as drastic as the error covariance divergence for position and velocity ($10^5, 10^4$). The sawtooth pattern occurs again during the times without COPOSIT measurements. Overall, the lack of COPOSIT measurements towards the end of the approach and landing creates high levels of uncertainty in the state estimation. Future work may include integrating a nadir camera that contains the landmarks and fiducials during the last portion of the approach and landing, potentially increasing the number of detections for COPOSIT and higher confidence in the state estimation. Another strategy could be to implement a vision-based landing technique with a nadir camera that has the helipad markings within the field of view throughout most of the landing phase [29].

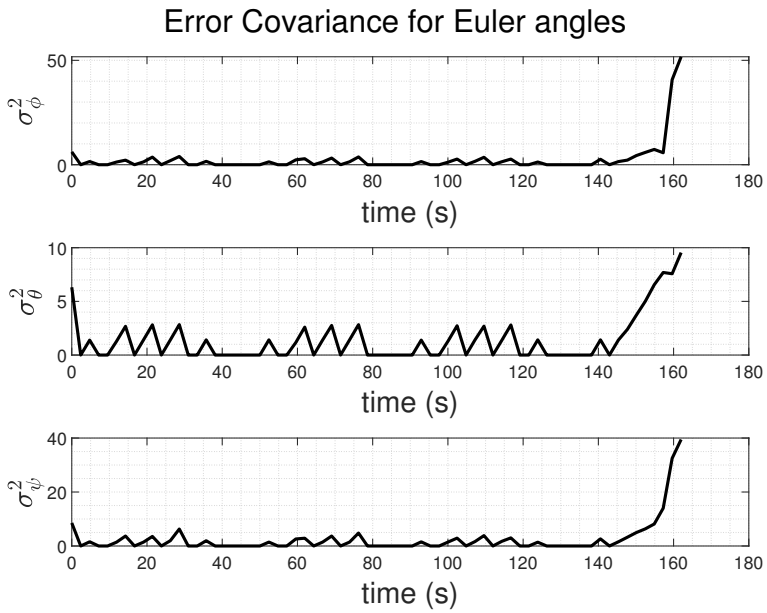


Fig. 23 Error Covariance: Euler Angles

4. State Estimation

The EKF in VAL computes velocity measurements based on the COPOSIT position measurements to yield higher accuracy in estimation since preliminary tests that neglected COPOSIT velocity measurements led to significant divergence [25]. Figure 24 shows the position and velocity state estimates, which fluctuate around the nominal states (black lines). Figure 25 demonstrates similar behavior but with more significant fluctuations in which the lack of detections significantly affects the Euler angle estimation, leading to substantial drift without COPOSIT measurements.

These fluctuations occur due to the lack of COPOSIT measurements but return to the nominal states when COPOSIT measurements become available. Thus, sawtooth patterns form, but drastic divergence occurs towards the end of the approach and landing from the lack of COPOSIT measurements, which arises due to the lack of key landmarks and fiducials (recall Table 2) in the field of view.

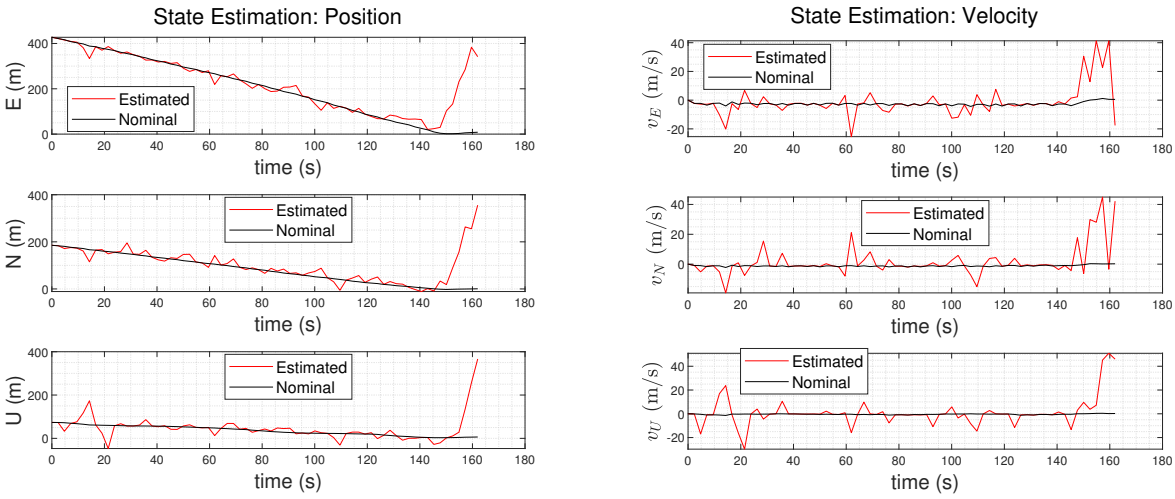
Tables 7 - 8 show the average and standard deviation of the error of the EKF with and without COPOSIT measurements, which demonstrates how significant COPOSIT has on the state estimation. The average and standard deviations are significantly larger (orders of magnitude) without COPOSIT measurements. Due to the constant drop-ins and outs of COPOSIT measurements, the overall position estimation accuracy is worse than the nominal GPS horizontal accuracy, which ranges from 2 to 4 m (see subsection VI.C). Overall, consistently detecting at least four of the landmarks and fiducials throughout the trajectory reduces the sawtooth pattern, leading to higher state estimation accuracy.

Table 7 State Estimation Statistics of the Error with COPOSIT Measurements

	E (m)	N (m)	U (m)	v_E (m/s)	v_N (m/s)	v_U (m/s)	ϕ (rad)	θ (rad)	ψ (rad)
μ	7.125	5.303	5.202	0.245	0.179	0.106	0.000754	0.00103	0.000742
σ	9.758	6.460	6.763	0.283	0.154	0.131	0.000700	0.00114	0.000903

Table 8 State Estimation Statistics of the Error without COPOSIT Measurements

	E (m)	N (m)	U (m)	v_E (m/s)	v_N (m/s)	v_U (m/s)	ϕ (rad)	θ (rad)	ψ (rad)
μ	59.776	53.001	45.483	10.257	10.646	12.478	0.474	0.351	0.339
σ	99.989	84.481	76.818	10.658	11.340	13.305	0.330	0.249	0.228



(a) State Estimation: Position **(b) State Estimation: Velocity**

Fig. 24 State Estimation: Position & Velocity

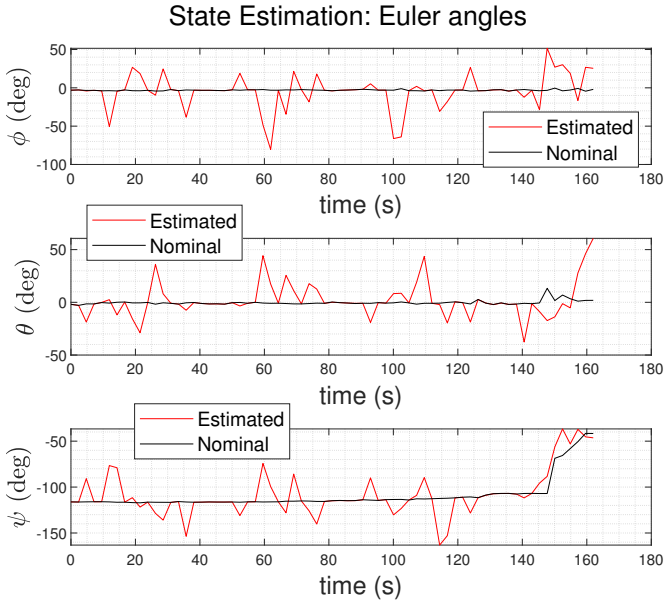
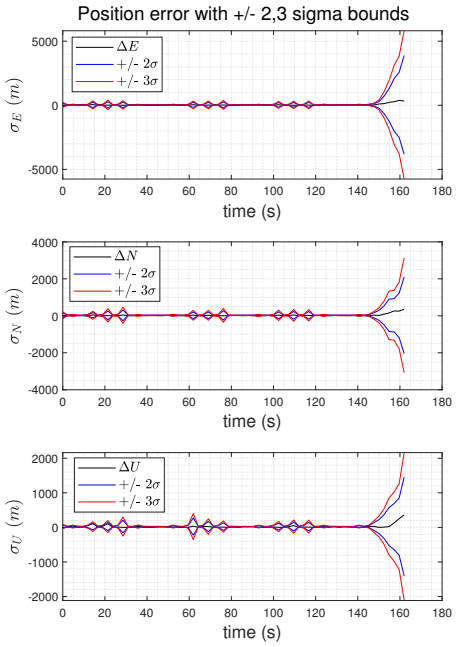


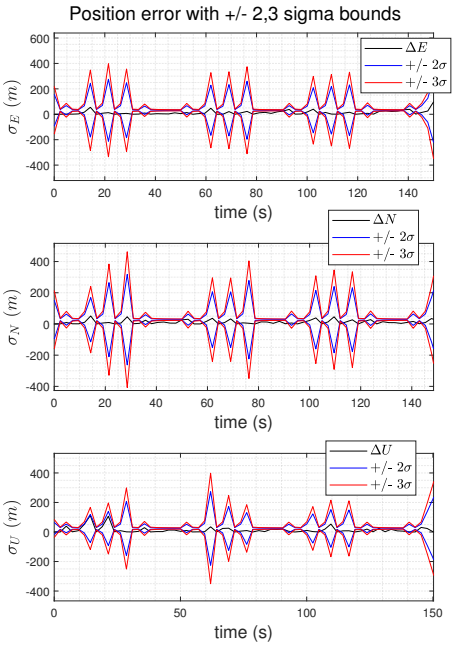
Fig. 25 State Estimation: Euler Angles

5. Bounded Error

Figures 26 - 28 show the position, velocity, and Euler angle errors with $\pm 2, 3\sigma$ centered around the mean error. The blue lines are the $\pm 2\sigma$ bounds, while the red lines are the $\pm 3\sigma$ bounds. Initial inspections imply that the errors are bounded well, but zooming in before the significant divergence at the end shows large spikes in the sawtooth pattern due to the availability of the COPOSIT measurements. These large amplitudes follow the significant mean and standard deviation of the errors shown earlier in Tables 7 - 8. Extreme divergence occurs at the end due to the lack of detections and COPOSIT measurements since all the landmarks and fiducials are outside the field of view.



(a) Error with $\pm 2, 3\sigma$ Bounds: Position

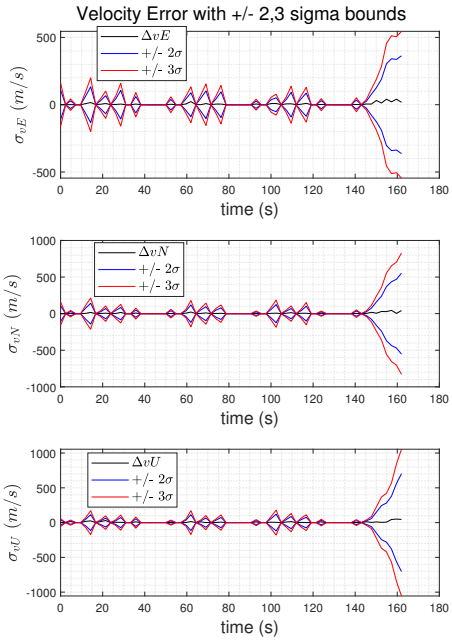


(b) Error with $\pm 2, 3\sigma$ Bounds: Position (Zoomed In)

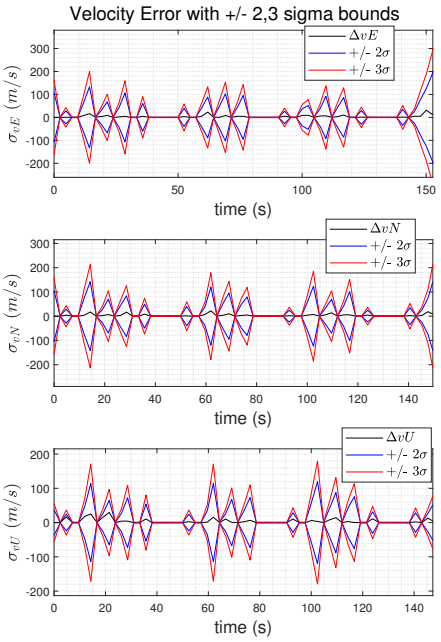
Fig. 26 Error with $\pm 2, 3\sigma$ Bounds: Position

Figure 29 compares the telemetry, ORB SLAM 2, and VAL EKF position vector magnitude. The VAL EKF position estimation tends to match with the telemetry data until the end due to the lack of detections and COPOSIT measurements, which causes divergence and high uncertainty in the EKF state estimation. Conversely, the ORB SLAM 2 position magnitude does not closely follow the telemetry data well until the end.

Overall, the VAL EKF position magnitude is accurate until the final moments of landing due to the lack of distributed landmarks and fiducials in the field of view. In contrast, ORB SLAM 2 position magnitude tends to become more accurate towards the end of the trajectory before landing. The VAL EKF works well when the camera has the distributed landmarks and fiducials in its field of view. At the same time, ORB SLAM 2 shows convergence towards the final moments before landing without the distributed landmarks and fiducials in the camera's field of view. Therefore, future work can consider combining ORB SLAM 2 or another VSLAM method with VAL EKF to obtain a best-of-both-worlds approach by including known and unknown a priori features in the scenery.

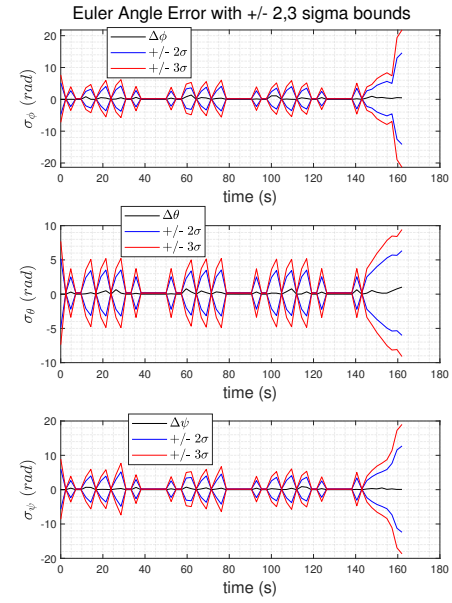


(a) Error with $\pm 2, 3\sigma$ Bounds: Velocity

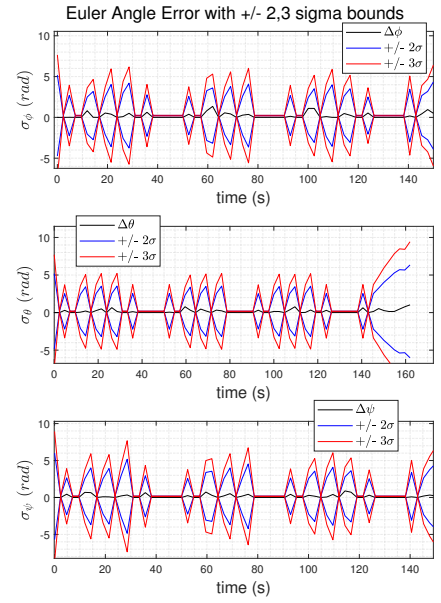


(b) Error with $\pm 2, 3\sigma$ Bounds: Velocity (Zoomed In)

Fig. 27 Error with $\pm 2, 3\sigma$ Bounds: Velocity



(a) Error with $\pm 2, 3\sigma$ Bounds: Euler angles



(b) Error with $\pm 2, 3\sigma$ Bounds: Euler angles (Zoomed In)

Fig. 28 Error with $\pm 2, 3\sigma$ Bounds: Euler angles

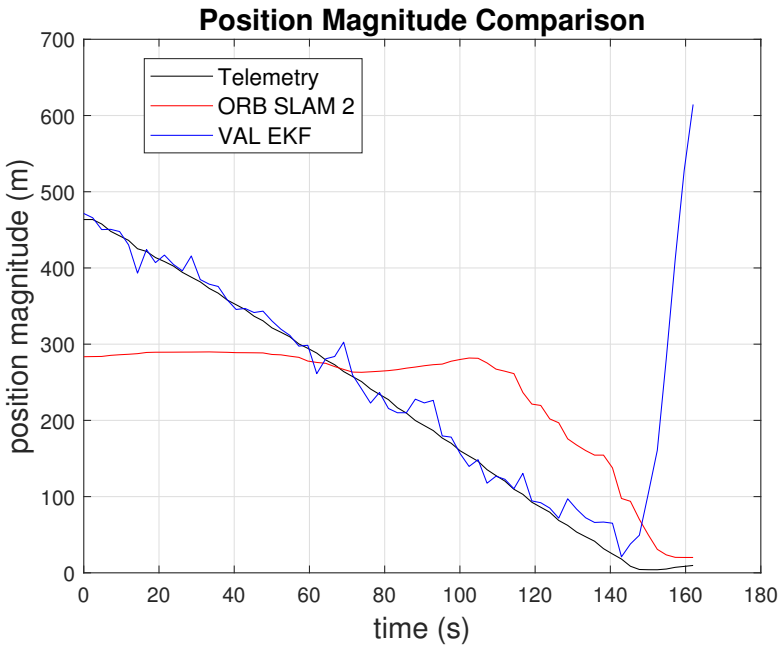


Fig. 29 Position Magnitude Comparison

VII. Conclusion

This initial comparative computer vision study for AAM approach and landing compares ORB SLAM 2 and the EKF navigation solution from Ref. [24] that combines onboard IMU measurements and coplanar POSIT. ORB SLAM 2 serves as a baseline VSLAM navigation method for comparing against the COPOSIT-EKF system as potential APNT solutions. However, it may yield inaccurate solutions for approach and landing when there are dynamic features such as birds or other aircraft. These different approaches deliver alternative onboard APNT solutions to GPS/GNSS-based methods, which is essential for flying in urban environments with degraded GPS signals. The COPOSIT-EKF has accurate estimation until the end due to the lack of distributed landmarks and fiducials within the camera's view, which causes divergence and drift. Conversely, the error in ORB SLAM 2 tends to decrease during the final moments before landing. Future work can combine both methods for a best-of-both-worlds APNT solution with a fusion of unknown and known feature points to yield accurate state estimation throughout the entire approach and landing profile. Conducting UAV flight tests with approach and landing at the NASA AFRC helipad provides experimental data, which serves as an excellent test case for simulating future AAM approach and landing. Future work can investigate whether feature detection and correspondence still produce accurate results at higher cruise speeds. Future flight tests with larger aircraft like helicopters and different flight conditions such as night, dawn, dusk, fog, and rain will provide additional insight for simulating and testing AAM approach and landing.

Acknowledgments

The authors are very grateful to the Dale Reed Subscale Flight Research Laboratory members at NASA Armstrong Flight Research Center for obtaining UAS experimental results to serve as ground truth for the APNT solutions presented in this paper. The authors would also like to thank the Transformative Tools and Technologies (TTT) Project under the NASA Aeronautics Research Mission Directorate (ARMD) for funding this research.

References

[1] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints," *International journal of computer vision*, Vol. 60, No. 2, 2004, pp. 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[2] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., "Speeded-up Robust Features (SURF)," *Computer vision and image understanding*, Vol. 110, No. 3, 2008, pp. 346–359. https://doi.org/10.1016/j.cviu.2007.09.014.

[3] Harris, C., and Stephens, M., "A Combined Corner and Edge Detector," *Alvey vision conference*, Vol. 15, No. 50, Citeseer, 1988, pp. 10–5244. URL http://www.eng.tau.ac.il/~cvapps/Supplement/[%201988%20]%20A%20Combind%20Corner%20Aand%20Edge%20Detector.pdf.

[4] Rosten, E., and Drummond, T., "Fusing Points and Lines for High Performance Tracking," *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2, IEEE, 2005, pp. 1508–1515. https://doi.org/10.1109/ICCV.2005.104.

[5] Oliveira, F. P., and Tavares, J. M. R., "Medical Image Registration: A Review," *Computer methods in biomechanics and biomedical engineering*, Vol. 17, No. 2, 2014, pp. 73–93. https://doi.org/10.1080/10255842.2012.670855.

[6] Yuen, H., Princen, J., Illingworth, J., and Kittler, J., "Comparative Study of Hough Transform Methods for Circle Finding," *Image and vision computing*, Vol. 8, No. 1, 1990, pp. 71–77. https://doi.org/10.1016/0262-8856(90)90059-E.

[7] Atherton, T. J., and Kerbyson, D. J., "Size Invariant Circle Detection," *Image and Vision computing*, Vol. 17, No. 11, 1999, pp. 795–803. https://doi.org/10.1016/S0262-8856(98)00160-7.

[8] Alcantarilla, P. F., Bartoli, A., and Davison, A. J., "KAZE Features," *European conference on computer vision*, Springer, 2012, pp. 214–227. https://doi.org/10.1007/978-3-642-33783-3_16.

[9] Shi, J., et al., "Good Features to Track," *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, IEEE, 1994, pp. 593–600. https://doi.org/10.1109/CVPR.1994.323794.

[10] Leutenegger, S., Chli, M., and Siegwart, R. Y., "BRISK: Binary Robust Invariant Scalable Keypoints," *2011 International conference on computer vision*, IEEE, 2011, pp. 2548–2555. https://doi.org/10.1109/ICCV.2011.6126542.

[11] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: An Efficient Alternative to SIFT or SURF," *2011 International conference on computer vision*, IEEE, 2011, pp. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544.

[12] Matas, J., Chum, O., Urban, M., and Pajdla, T., "Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions," *Image and vision computing*, Vol. 22, No. 10, 2004, pp. 761–767. https://doi.org/10.1016/j.imavis.2004.02.006.

[13] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D., "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE transactions on robotics*, Vol. 31, No. 5, 2015, pp. 1147–1163. https://doi.org/10.1109/TRO.2015.2463671.

[14] Mur-Artal, R., and Tardós, J. D., "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE transactions on robotics*, Vol. 33, No. 5, 2017, pp. 1255–1262. https://doi.org/10.1109/TRO.2017.2705103.

[15] Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., and Tardós, J. D., "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, Vol. 37, No. 6, 2021, pp. 1874–1890. https://doi.org/10.1109/TRO.2021.3075644.

[16] Zhao, Y., and Vela, P. A., "Good Feature Selection for Least Squares Pose Optimization in VO/VSLAM," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1183–1189. https://doi.org/10.1109/IROS.2018.8593641.

[17] Won, D. H., Chun, S., Sung, S., Lee, Y. J., Cho, J., Joo, J., and Park, J., "INS/vSLAM System Using Distributed Particle Filter," *International Journal of Control, Automation and Systems*, Vol. 8, No. 6, 2010, pp. 1232–1240. https://doi.org/10.1007/s12555-010-0608-7.

[18] Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., and Shein, A., "GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft," *Journal of field robotics*, Vol. 30, No. 3, 2013, pp. 415–438. https://doi.org/10.1002/rob.21454.

[19] Yu, L., Yang, E., and Yang, B., "AFE-ORB-SLAM: Robust Monocular VSLAM Based on Adaptive FAST Threshold and Image Enhancement for Complex Lighting Environments," *Journal of Intelligent & Robotic Systems*, Vol. 105, No. 2, 2022, pp. 1–14. https://doi.org/10.1007/s10846-022-01645-w.

[20] Klippenstein, J., and Zhang, H., "Performance Evaluation of Visual SLAM Using Several Feature Extractors," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 1574–1581. https://doi.org/10.1109/IROS.2009.5354001.

[21] Wang, T., Wang, C., Liang, J., and Zhang, Y., "Rao-Blackwellized Visual SLAM for Small UAVs with Vehicle Model Partition," *Industrial Robot: An International Journal*, 2014. https://doi.org/10.1108/IR-07-2013-378.

[22] Latcham, P., and Taylor, C. N., "Comparison of Visual Simultaneous Localization and Mapping Methods for Fixed-Wing Aircraft Using SLAMBench2," *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1578–1586. https://doi.org/10.1109/PLANS46316.2020.9109945.

[23] Cui, T., Guo, C., Liu, Y., and Tian, Z., "Precise Landing Control of UAV Based on Binocular Visual SLAM," *2021 4th International Conference on Intelligent Autonomous Systems (ICoIAS)*, IEEE, 2021, pp. 312–317. https://doi.org/10.1109/ICoIAS53694.2021.00062.

[24] Kawamura, E., Kannan, K., Lombaerts, T., and Ippolito, C. A., "Vision-Based Precision Approach and Landing for Advanced Air Mobility," *AIAA SciTech 2022 Forum*, AIAA 2022-0497, 2022. https://doi.org/10.2514/6.2022-0497.

[25] Kawamura, E., Dolph, C., Kannan, K., Lombaerts, T., and Ippolito, C. A., "Simulated Vision-based Approach and Landing System Advanced Air Mobility," *AIAA SciTech 2023 Forum*, AIAA-2023-2195, 2023. https://doi.org/10.2514/6.2023-2195.

[26] Thompson, N., "NASA National Campaign Build 1, Edwards AFB, California," National Geospatial-Intelligence Agency, 2020.

[27] Webber, D., and Zahn, D., "FAA and the National Campaign," [Powerpoint], 2021.

[28] Oberkampf, D., DeMenthon, D. F., and Davis, L. S., "Iterative Pose Estimation Using Coplanar Feature Points," *Computer Vision and Image Understanding*, Vol. 63, No. 3, 1996, pp. 495–511. https://doi.org/10.1006/cviu.1996.0037.

[29] Ye, S., Wan, Z., Zeng, L., Li, C., and Zhang, Y., "A Vision-Based Navigation Method for eVTOL Final Approach in Urban Air Mobility (UAM)," *2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, IEEE, 2020, pp. 645–649. https://doi.org/10.1109/CVCI51460.2020.9338487.