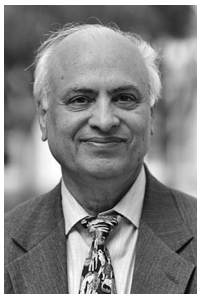
Refinement and Characterization of Feature-Based Vision Algorithm for Ship-Deck Landing under Degraded Visibility



Victoria Britcher*
Clark Doctoral Fellow

Anubhav Datta
Alfred Gessow Professor

Inderjit Chopra
Clark Professor and Director

Alfred Gessow Rotorcraft Center, University of Maryland, College Park, MD

The objective of this work is to refine and experimentally characterize a two-dimensional, feature-based vision algorithm for tracking a stochastically moving ship-deck under degraded visual conditions. A 2.75-kg quadrotor UAV (unmanned aerial vehicle), with only the accessories essential for vision-based navigation, is specifically designed to establish the minimal system requirements, fabricated, and tested in-house. The algorithm was integrated into this quadrotor and was performance tested in controlled, hand-held tests as well as piloted free-flight hover. All results were validated against Vicon ground-truth data. In the controlled tests, the algorithm was first used to estimate the motion of a Stewart platform simulating a stochastically moving ship-deck. Next, tests were conducted under visually degraded conditions, specifically glare, low illumination, and occlusion of the landing pad. Free-flight tests were conducted with the quadrotor hovering above the landing pad at varying levels of illumination and occlusion as well as with ship-deck motion. The algorithm could accurately estimate the pose of a ship-deck undergoing Sea-state 6 motions in visually degraded conditions in both hand-held and free-flight tests. Performance was observed to reduce slightly in free-flight compared to hand-held tests due to aircraft motion and vibration.

Nomenclature

v_k	observation or measurement noise in Kalman filter
w_k	process noise in Kalman filter
x_k, y_k, z_k	position at time k
$\dot{x}_k, \dot{y}_k, \dot{z}_k$	derivative of position at time k

Abbreviations

ESC	electronic speed controller
IMU	inertial measurement unit
RMSE	root mean square error
ROS	Robot Operating System
SIFT	scale invariant feature transform
UAS	uncrewed aerial system
UAV	unmanned aerial vehicle

Introduction

Landing a helicopter on a moving ship-deck is one of the most challenging and dangerous tasks for a human pilot. Therefore, the overall objective of this research is to work towards developing autonomous, vision-based ship-deck landing, which will significantly improve safety for pilots.

Many previous studies have addressed the problem of vision-based rotorcraft landing, with most focusing on multirotor UAVs (unmanned aerial vehicles) landing on stationary or linearly moving targets. Several approaches have been explored for detecting the landing target. This includes fiducial markers or tags, such as ArUco markers (Refs. 1–3), simple custom-designed tags (Refs. 4, 5), or lights or markings on the landing pad (Ref. 6). Feature detection and matching have also been used to detect known areas of ground terrain near a landing location (Ref. 7). Recently, machine learning approaches have been investigated (Ref. 8).

Some research has investigated the specific problem of landing on ship-decks using vision. Ship-decks produce large, stochastically oscillating motions, introducing significant difficulties to both visual tracking and safe landing. Similarly to the more general work on rotorcraft landing, previous work on ship-deck landing has mainly utilized simple visual references to detect ship-deck motion. This includes fiducial markers (Refs. 9, 10), H-shaped landing targets (Refs. 11, 12), and arrangements of lights or targets (Refs. 13–15). Detecting ship structures using machine learning techniques has been explored in more recent work (Refs. 16, 17).

Our previous work investigated the use of vision for ship-deck landing, particularly for tracking stochastic ship-deck motion in high sea states, where severe oscillations are present. As described in our conference paper (Ref. 18), it was validated that a ship-deck could be detected and tracked using only vision with a single camera. A quadrotor was developed in-house to demonstrate tracking and landing on a stochastically moving ship-deck. A fiducial-based algorithm, which detects the pose (position and orientation) of an AprilTag marker on the ship-deck, was used and executed on the quadrotor using an onboard flight computer. In experimental tests, the algorithm was able to detect and track the ship-deck at various oscillation frequencies as well as for stochastic

*Corresponding author; email: vbritche@umd.edu. Manuscript received March 2024; accepted October 2024.

 © 2025 Vertical Flight Society

motion. However, the fiducial-based algorithm fails in visually degraded conditions, including for any type of occlusion on the marker, since the entire marker must be clearly visible for detection and pose estimation.

To rectify this deficiency, a new feature-based vision system was developed gradually as reported in our conference papers (Refs. 19, 20). While feature detection has been investigated for rotorcraft landing (Ref. 7), this technique has not been previously used to track the pose of a moving landing platform. This new algorithm is intended to accurately track the pose of a ship-deck in both ideal and visually degraded conditions. It was verified to be robust to occlusions and illumination differences under simple bench-top test conditions. This algorithm is also capable of working with generic landing pad patterns, unlike the fiducial-based algorithm, which requires a predesigned tag. However, the feature-based algorithm was not previously implemented on an actual aircraft. It is also significantly more computationally intensive than fiducial detection algorithms, requiring a more powerful and physically larger onboard flight computer.

This paper documents the final feature-based algorithm and its experimental validation. First, the development of a larger quadrotor UAV incorporating the new flight computer is described. Next, we provide the results of controlled algorithm testing for ship-deck motion and under visually degraded conditions. Finally, the algorithm performance is evaluated for free flight, in visually ideal and visually degraded scenarios.

Quadrotor Development

The vision algorithm must function using only onboard computer and vision hardware, as ship-deck landings may occur in isolated locations or poor weather, where external communication is unreliable. The feature-based vision algorithm is computationally intensive, requiring a powerful onboard computer to operate in real time. However, due to physical size and weight constraints, it was not possible to integrate the required computer into the older quadrotor used for the demonstration of our fiducial-based algorithm. Additionally, due to the need to customize the onboard vision and computer hardware based on research requirements, developing a custom quadrotor in-house was necessary over utilizing an off-the-shelf UAV. As a result, we have developed a much larger quadrotor UAV sized, designed, and fabricated around this new computer and other ancillary hardware.

Flight computer

While the feature-based algorithm is more capable than our previous fiducial-based system, it is also much more computationally intensive. An UP Xtreme embedded computer, featuring an Intel i5 processor, was selected for the large quadrotor UAV.

This computer is able to update the vision algorithm at a maximum rate of 12 Hz in benchtop tests. In contrast, the computer used in previous experiments is only capable of updating the feature-based algorithm at a rate of 0.7 Hz, far too slow for real-time operation (Ref. 19).

Structure and specifications

The new, larger quadrotor was developed and fabricated in-house using a combination of in-house custom-built and off-the-shelf components. The overall specifications of the final aircraft are listed in Table 1. A CAD model of the quadrotor was first developed as shown in Fig. 1, with the overall layout and major components. The mass and moments of inertia were estimated from CAD software. In designing and choosing components for this UAV, we kept in mind future modifications and possible hardware additions. Therefore, the quadrotor was designed to be larger and to carry a greater payload than the current test hardware.

Table 1. Quadrotor characteristics

Parameters	Value
Gross takeoff mass	2.75 kg
Rotor spacing (x and y)	0.45 m
Rotor diameter	0.229 m
Rotor RPM at hover	7300 RPM
Center of mass (x)	−0.020 m
Center of mass (y)	0.0 m
Ixx	0.0350 kgm^2
Iyy	0.0699 kgm^2
Izz	0.1018 kgm^2
Battery mass	1.0 kg
Hover endurance	12.8 min

This allows components to be added in the future with fewer weight or geometric constraints.

The new quadrotor has a frame size of 450 mm × 450 mm (measured from the centers of the motors) and a gross take-off weight of 2.75 kg. This is much larger than the previous quadrotor, which had a frame size of 270 mm × 270 mm. The frame is constructed of 0.5 inches (12.7 mm) 6061-T6 rectangular aluminum tubes with 1/16 inch (1.59 mm) wall thickness. The 6061 aluminum alloy offers greater strength compared to the 6063 aluminum alloy used in the previous quadrotor. The tubes are arranged in an H-shape and attached with steel L-shaped corner brackets, similar to the configuration used in the previous quadrotor. Aluminum tubes were chosen because they provide structural strength and stiffness while being easy to machine.

The key components, namely the flight computer, flight controller, batteries, and camera, are mounted to laser-cut acrylic plates mounted to the frame. If any components need to be changed or rearranged, or if these acrylic plates are damaged in a crash, these items can be quickly re-cut with in-house laser cutters and replaced.

The motor mounts and landing gear were also custom-designed and produced using in-house additive manufacturing facilities. They were printed using a Markforged Mark 2 three-dimensional (3D) printer using Onyx, a micro carbon fiber filled nylon material. Additional strength was provided through printed carbon-fiber reinforcement layers.

A photograph of the assembled quadrotor is shown in Fig. 2.

Propulsion and power

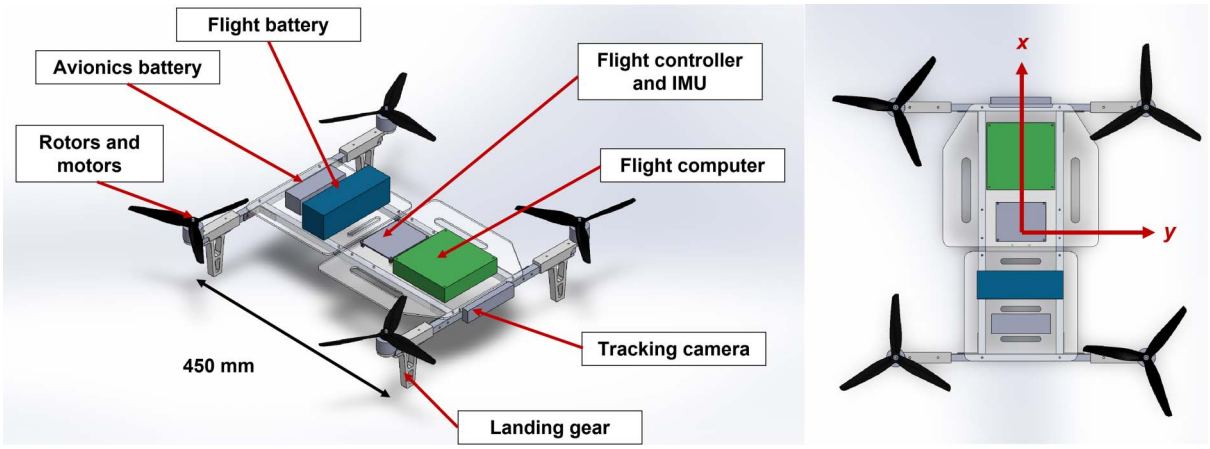
The vehicle features four 9-inch diameter, three-bladed rotors, driven by T-Motor Cine66 925KV 2812 brushless electric motors. Using manufacturer-provided test stand data for this motor/rotor combination, we calculated a maximum hover endurance of 12.8 min.

The four motors are driven by a Holybro Tekko32 4-in-1 electronic speed controller (ESC). This ESC supports a maximum current of 65 A for each motor. While the motor current is typically well below this level, in the case of sudden maneuvers or transients, this provides a factor of safety above the maximum motor current provided in test data.

Separate lithium polymer batteries are used to power the motors and avionics. A 1500 mAh, four cell (14.8 V), 120C battery powers the avionics, including the flight computer. Another 6000 mAh, six cell (25.2 V), 50C battery is used to power the motors via the ESC.

Avionics and control

The flight control loop used for attitude control and stabilization is executed on a Teensy 4.0 microcontroller with an ARM Cortex-M7 processor, which is capable of updating the attitude control loop at 1000 Hz.



(a) Major components (b) Axes system

Fig. 1. CAD diagram of new quadrotor showing major components and axes system. Note that the z-axis is positive down.

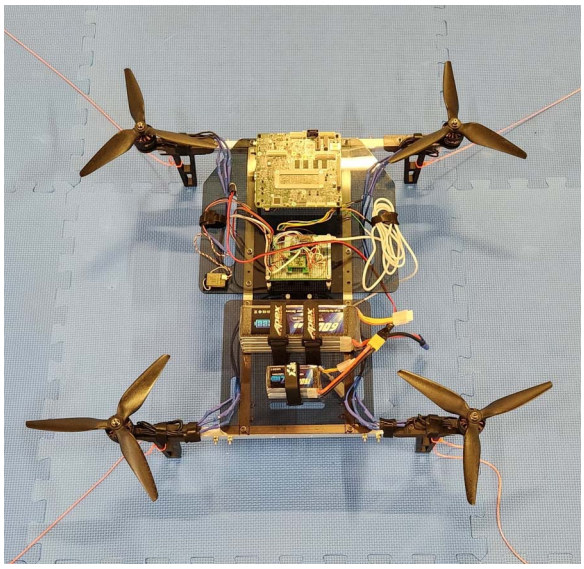


Fig. 2. Photo of the new larger quadrotor.

A ST LSM6DSOX inertial measurement unit (IMU), incorporating a gyroscope and accelerometer, is used to measure angular velocity and linear acceleration along three axes, respectively. The Teensy communicates with the IMU through the I2C communication protocol and collects updated measurements on each control loop update. The Teensy then sends commands for each motor to the ESC using the digital DShot protocol.

The flight controller receives commands from a 2.4-GHz radio receiver using a universal asynchronous receiver and transmitter, a serial communication method, utilizing an open-source library to read each message. This receiver is paired with a transmitter, enabling an operator to manually control the quadrotor.

The flight controller stabilizes attitude with nested proportional-integral-derivative controllers. The inner control loop takes in desired angular rates and outputs desired moments. These moments are converted to motor commands using a control mixer. Gyroscope measurements are used to calculate the current angular rates. A Kalman filter is applied to the raw gyroscope measurements before sending to the controller in order to filter noise and produce more reliable angular rate estimates. The outer control loop takes in desired attitude angles and outputs desired angular rates, which are sent to the inner loop controller. A Kalman filter fuses measurements from the accelerometer and gyroscope to produce estimated attitude angles.

Vision hardware

Images of the ship-deck are collected using a See3CAM USB camera, manufactured by e-con Systems. This is the same camera previously used for initial benchtop testing of the feature-based algorithm. The camera is mounted on the underside of the quadrotor, facing downward. This camera features a built-in autoexposure algorithm, which automatically varies the camera shutter speed for varying light intensity received by the camera. This allows the camera to compensate for a wide range of scene illumination, though at the cost of increased motion blur for low illumination levels.

Vision Algorithm

In this paper, we experimentally validate the performance of the two-dimensional (2D) feature-based algorithm we developed earlier in Ref. 19. The algorithm is briefly described here for completeness.

For this work, functions provided in OpenCV, an open-source computer vision library, were used for various image-processing tasks, such as image reading and scaling. Additionally, we utilized feature detection and matching functions provided in OpenCV.

Feature extraction and matching

Features, in this context, can be described as small patches of interest in an image. Features should preferably be invariant to transformations in scale, translation, in- and out-of-plane rotation, and illumination.

Locations of interest in an image are typically found by detecting strong corners or blobs, which have significant intensity variation in both directions. Image patches at these locations are extracted and converted to feature descriptors, which are a numerical representation of the image patch. Extracted features can be matched to find corresponding points across multiple images.

SIFT algorithm

The scale-invariant feature transform (SIFT) algorithm, developed by Lowe (Ref. 21), is a commonly used algorithm for feature detection and

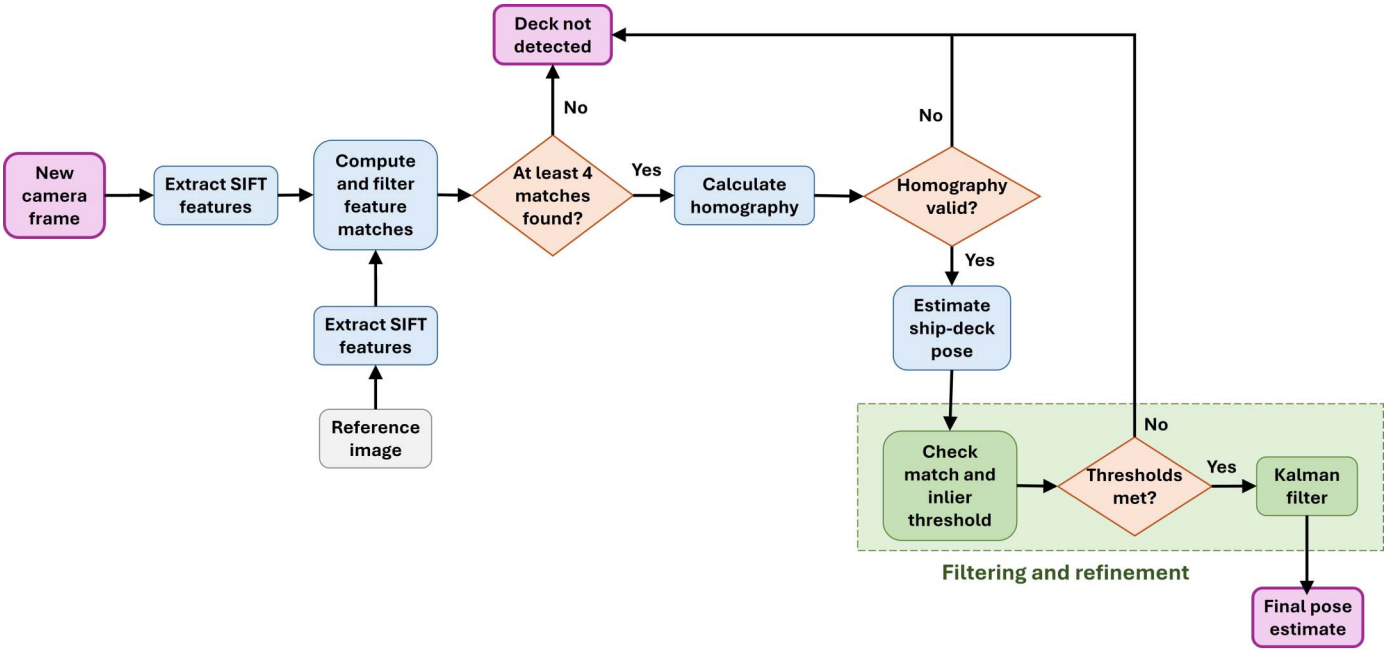


Fig. 3. Flowchart illustrating the steps in the vision algorithm.

extraction. The algorithm generates vectors of length 128 as feature descriptors. The algorithm is invariant to changes in scale and translation and is somewhat resilient to changes of in- and out-of-plane rotation and illumination.

Good SIFT features should be distinctive, meaning that these features can be reliably matched across multiple images with few incorrect matches. Good features typically consist of corner- or blob-like regions with high contrast. Low contrast or nondistinctive features are undesirable for feature-matching tasks.

For this work, we utilized an implementation of SIFT provided in OpenCV.

Ship-deck detection algorithm

The present algorithm detects and extracts SIFT features in images of the moving ship-deck. These features were matched to features extracted from a reference image of the landing pad.

The following procedure was used to detect and estimate the pose of the ship deck:

1) First, extract SIFT features from a reference image of the pattern located on the ship-deck. Any image may be used with the algorithm, but an image with many good SIFT features is preferable.

2) Load the most recent frame from the quadrotor camera and extract SIFT features from the image.

3) For each feature in the reference image, find the two closest matches (best and second best match) in the features extracted from the camera image. The distance between features is found by calculating the L2-norm of the difference between the feature descriptors.

4) Filter out poor matches using a series of tests: (1) *Lowe's ratio test*: the ratio of the calculated distance for the best and second best match must be below a threshold, 0.8 in this case (Ref. 21); (2) *Symmetry test*: the best feature match from the reference image to the camera frame should also be the best match from the camera frame to the reference image; (3) *Distance test*: the distance to a matched feature should be less than the minimum distance across all matches multiplied by a constant, 3 in this case. A match that fails any of these tests will be removed as a bad match.

5) A minimum of four matches are required to calculate a homography, or linear transformation, from the reference image to the camera image. If at least four feature matches are found after filtering poor matches, proceed with the following steps. Otherwise, report that the deck was not detected.

6) Use the Random Sample Consensus (RANSAC) algorithm to find a good homography. This algorithm works by repeatedly taking random samples of the matched points, which are used to calculate a homography. The remaining matched points are checked for consistency with the estimated homography. This procedure produces a result that is robust to outliers (Ref. 22).

7) Use the homography to transform the corners of the deck image. Verify that the transformed points are geometrically valid by checking that the transformed deck corners form a convex quadrilateral. If this is the case, then compute the pose of the deck, or its position and orientation in 3D relative to the camera (Ref. 23).

Algorithm implementation

Figure 3 illustrates the processing steps in the vision algorithm. The filtering and refinement steps, applied to the initial pose estimate obtained using the procedure discussed above, are discussed in the next section.

The feature-based vision algorithm was written in Python and implemented on the quadrotor for real-time operation. As discussed earlier, the OpenCV library was used for basic computer vision functions. It used ROS (robot operating system) 2, a set of robotics software libraries, for asynchronous messaging between individual processes, or nodes. The vision algorithm and logging runs on a single node. A separate node continuously retrieves frames from the camera and publishes them on a topic or a bus, allowing communication between nodes. The vision node subscribes to this topic, collecting updated camera frames as needed. Using ROS, the vision node can then publish the pose estimates, making this data available to other nodes such as a ship-deck tracking controller.

Algorithm Validation

Before this vision algorithm can be used for autonomous tracking and landing, it is necessary to validate the algorithm's performance and reliability under varying conditions. For validation, hand-held and free-flight hover tests of the vision algorithm were conducted.

A primary motivation for the feature-based algorithm is its robustness to visually difficult conditions, compared to fiducial-based approaches, where even minor visual degradation will result in detection failure. The algorithm should provide usable estimates of the pose of the landing target under visually degraded conditions; because of this, it is necessary to investigate conditions where the performance of the feature-based algorithm may deteriorate. Data was collected for visually ideal conditions—in this context, typical indoor lighting and no added occlusions or lighting limiting visibility of the landing image—as well as under various types of visual degradation.

First, several large-angle controlled tests, in which the quadrotor was held by hand, were carried out for ideal and visually degraded conditions. Next, tests were conducted in free-flight hover. Ground truth data was collected using a Vicon motion-capture system, which tracked the true pose of the quadrotor and landing target during tests.

For all tests, quadrotor yaw was held constant. Only minimal yaw motion was present in the ship-deck motion time histories. Due to the lack of yaw motion, and the greater importance of linear, roll, and pitch motions in landing, yaw results are omitted from result plots.

In all tests, images from the camera were scaled down by one-half from their original resolution of 1920×1080 pixels and were processed by the vision algorithm at a resolution of 960×540 pixels. This was needed because feature extraction is much more computationally expensive at higher resolutions, causing the algorithm to update too slowly for real-time operation.

It is important to note that the vision algorithm calculates the pose of the landing pad image relative to the quadrotor camera. This means that the relative pose is affected by both the movement of the quadrotor and the landing pad. Vicon ground truth data is also provided here as a relative pose, which is extracted from the true poses of the quadrotor and landing target in the world frame.

Initial Algorithm Validation and Refinement

Before evaluating more complex scenarios, simple initial hand-held tests were completed to provide baseline performance and uncover basic limitations in the algorithm, such as time delays. Though the feature-based algorithm can function in visually degraded scenarios, such conditions may potentially cause errors in the pose estimates from the algorithm. Therefore, this section also discusses techniques to refine the algorithm output.

Time delays and update frequency

From benchtop testing, a time delay of approximately 400 ms was observed in the timestamps of logged pose estimates compared to ground truth. This was further verified by examining logged data from hand-held testing. The delay is caused by a combination of two factors: (1) the computational time needed to compute a pose estimate and (2) latency in retrieving and publishing frames from the camera. The length of the delay may be reduced with improved hardware or through software optimizations, but cannot be entirely eliminated, and therefore must be accounted for during real-time tracking and landing.

As the time delay is consistent, the plotted results in this paper are shown with the pose estimate times shifted to remove the delay. This allows for better visual comparison in plots. Additionally, this allows us to quantitatively measure error using methods such as RMSE (root mean square error). Figure 4 shows an example of vision data with the delay present.

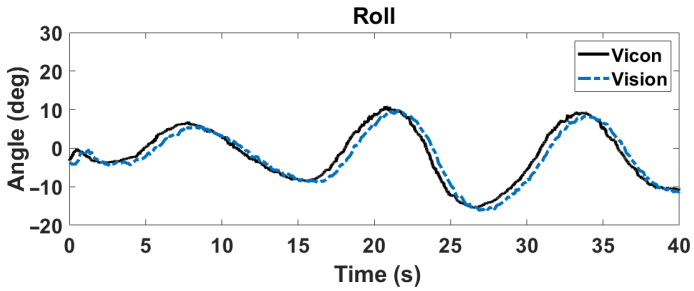


Fig. 4. Example of vision algorithm result for large roll motions with uncorrected 400 ms time delay.

The vision algorithm is computationally expensive, resulting in long delays between updates. In these tests, the algorithm demonstrated an update speed of approximately 9–11 Hz with the deck in frame. This was found to be adequately fast to estimate ship-deck oscillation, which displayed a maximum frequency of 0.34 Hz in the Navy SCONE dataset (Ref. 24). However, the low frequency of algorithm updates makes poor individual pose estimates problematic without additional filtering.

Baseline conditions

The algorithm's performance was evaluated under ideal, static conditions. This verified that the algorithm was producing the expected results and provided a baseline level of performance.

The test was conducted under normal indoor lighting (300 lux) and with the landing pad secured to a fixed, horizontal position on the floor. The quadrotor was held stationary above the landing pad, with the image fully in view of the camera, for several seconds.

The attitude and position estimation results are shown in Figs. 5 and 6. The estimates of both linear position and attitude had minimal error. The final RMSE of the linear position is below 1.3 cm on all axes. Roll, pitch, and yaw have an RMSE of 0.8, 1.4, and 1.5 deg, respectively.

Reference image selection

The feature-based algorithm can be used with any generic reference image as long as a sufficient number of features can be reliably detected and matched. This means that the existing markings on a ship-deck helipad could potentially be used as a reference image. However, these markings have relatively few good features compared to a custom, detailed reference image.

Algorithm performance was compared using two different reference images. The first, shown in Fig. 7(a), consists of a grid of AprilTag markers. This pattern was selected as it is rich in distinct, high-contrast features. The second image, shown in Fig. 7(b), is representative of typical helicopter landing pad markings.

The test was conducted under normal indoor lighting (300 lux). The quadrotor was held stationary above a landing pad with linear and attitude motion, keeping the image fully within the camera frame. The same motion trajectory was used for both images for consistency.

The RMSE results, given in Table 2, indicate a significant increase in pose estimation accuracy with a detailed, customized landing pad image compared to typical landing pad markings. This difference can be explained with the greater number of strong, distinctive features in the detailed image compared to the typical landing pad. This results in a greater number of feature matches between the reference and camera

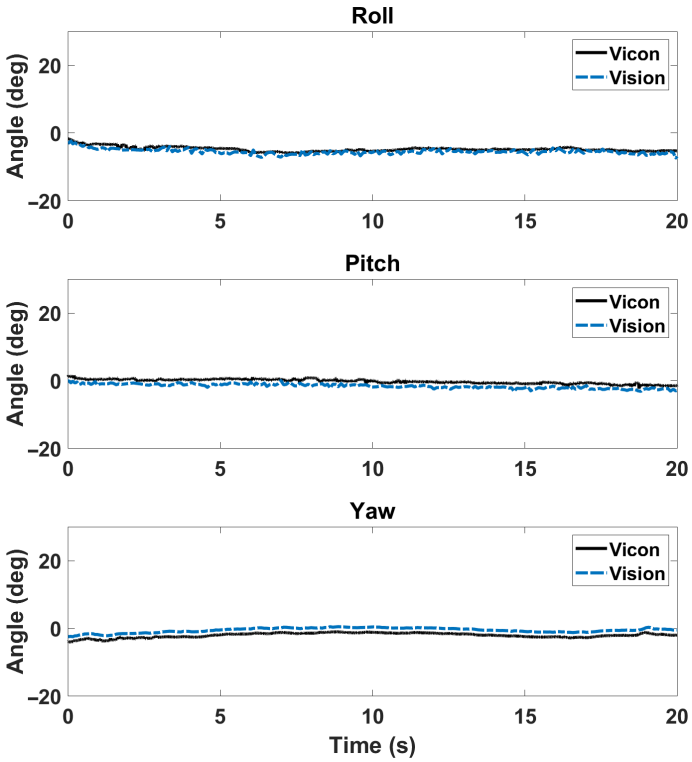


Fig. 5. Estimation of aircraft attitude angles relative to a stationary landing marker from onboard camera.

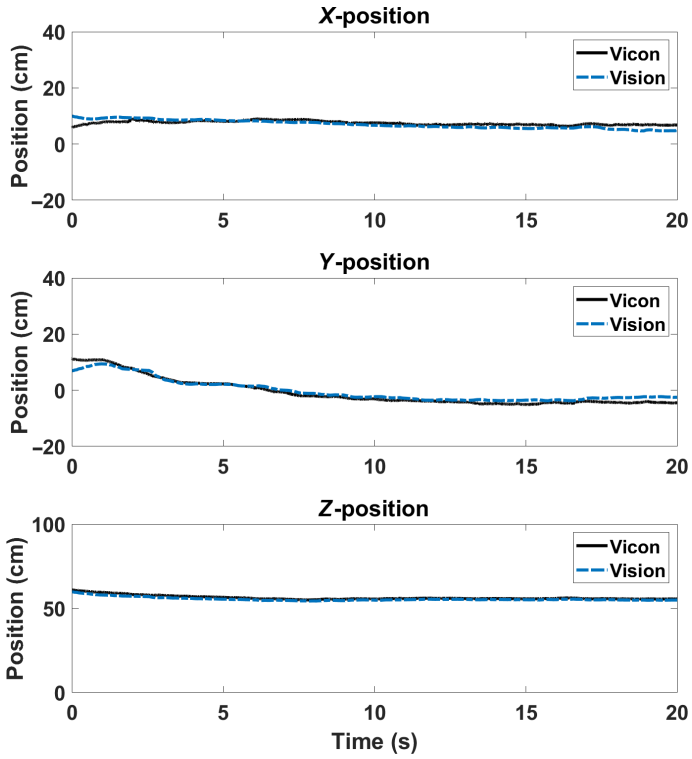


Fig. 6. Estimation of aircraft position relative to a stationary landing marker from onboard camera.

Table 2. RMSE for different reference images.

Test Case	*X* (cm)	*Y* (cm)	*Z* (cm)	Roll (deg)	Pitch (deg)
Customized image	0.4	0.5	1.2	1.6	1.6
Typical landing image	1.6	2.0	2.0	2.2	5.0

image. With the customized image, an average of 369 feature matches were obtained in each update; however, for the helipad markings, only 40 matches were obtained on average.

These results show that, while it is feasible to use a feature-matching algorithm with existing ship-deck markings, a customized image with many good features is ideal for good algorithm performance. The feature-rich image evaluated here is, therefore, used in all subsequent tests in this paper.

Feature match and inlier thresholds

Calculating a homography requires a minimum of four feature matches. However, with a small number of feature matches, incorrect matches can cause the calculated homography to be incorrect. When executing RANSAC, the number of inliers, or feature matches consistent with the homography, is calculated. The number of inliers is also useful to determine the quality of a pose estimate; a small number of inliers may indicate a poor result. The algorithm output could, therefore, be improved by implementing minimum thresholds for both the number of good feature matches and the number of inlier matches and rejecting images that fall below these thresholds.

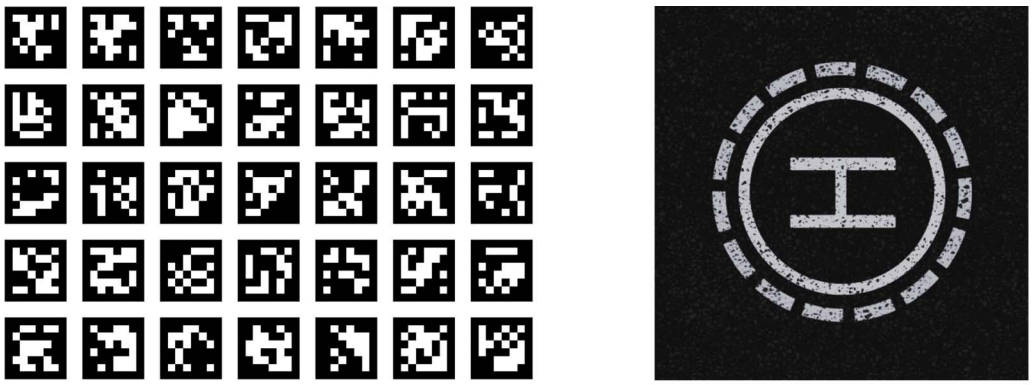
A vision algorithm test was completed to determine good match and inlier thresholds. Algorithm output was collected with the quadrotor held by hand above a stationary landing pad. The quadrotor was moved in quick, random motions in attitude and position, with the landing pad either fully or partially in view of the camera. For each pose estimate, the number of feature matches and inliers were logged. This was intended to generate pose estimates with varying numbers of matches and inliers.

The attitude and position results are shown in Figs. 8 and 9. The raw pose estimates with no thresholds (raw data) correspond to the output of the previously developed vision algorithm (Ref. 19). These results are plotted alongside data which is then filtered to remove pose estimates falling below match and inlier thresholds (thresholding). The unmodified vision algorithm produces a number of inaccurate and noisy pose estimates, particularly in roll and pitch; this inaccuracy was also observed during experimental testing in visually degraded conditions (Ref. 20). However, these poor estimates are removed with the added thresholding. The ideal minimum threshold for feature matches was found to be 25, while the best threshold for inliers was found to be 17. These values were chosen to reject a majority of inaccurate pose estimates while minimizing false rejections.

As shown in these results, applying feature match and inlier thresholds improves the pose estimates significantly, removing almost all of the poor estimates. These thresholds were incorporated into the vision algorithm and were applied to all tests later in this paper.

Kalman filters

A Kalman filter was also integrated into the vision code, providing filtered estimates of the ship-deck pose alongside the unfiltered estimates. The filter can mitigate potential noise and errors in the algorithm output with minimal computational cost.



(a) (b)

Fig. 7. Reference images evaluated using vision algorithm. (a) Detailed reference image composed of an AprilTag marker grid. (b) Image representative of typical ship-deck landing pads.

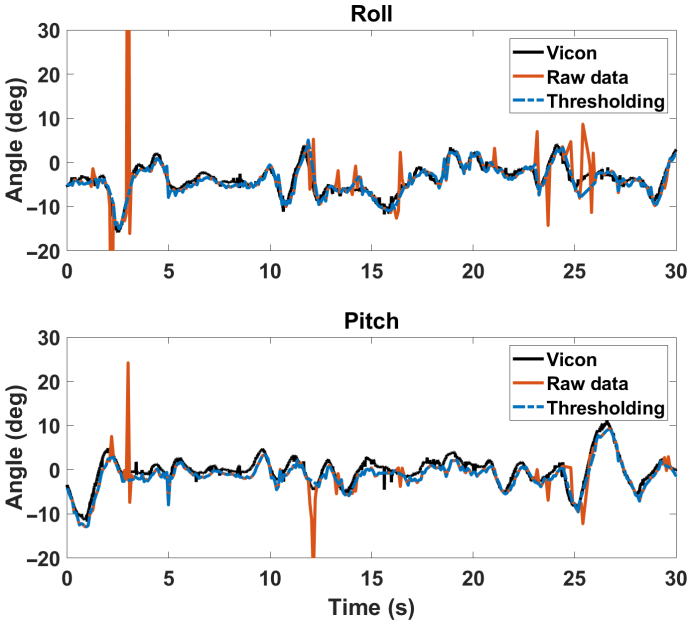


Fig. 8. Estimation of aircraft attitude angles relative to a stationary landing marker from onboard camera. Results are compared with and without the application of feature match and inlier thresholding.

The logged data for linear positions (X, Y, and Z positions) was processed using a 3D constant-velocity Kalman filter, shown below.

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ y_{k+1} \\ \dot{y}_{k+1} \\ z_{k+1} \\ \dot{z}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \\ z_k \\ \dot{z}_k \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 & 0 \\ \Delta t & 0 & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \frac{1}{2}\Delta t^2 \\ 0 & 0 & \Delta t \end{bmatrix} w_k \quad (1)$$

where x_k, y_k, and z_k are the linear position at timestep k. A separate Kalman filter, using the same constant-velocity model, was also used to process the estimated Euler angles (roll, pitch, and yaw).

The Kalman filter was integrated into the vision algorithm directly, filtering raw position and attitude estimates produced from ship-deck tracking. The code logs both raw and filtered pose estimates, allowing both outputs to be evaluated.

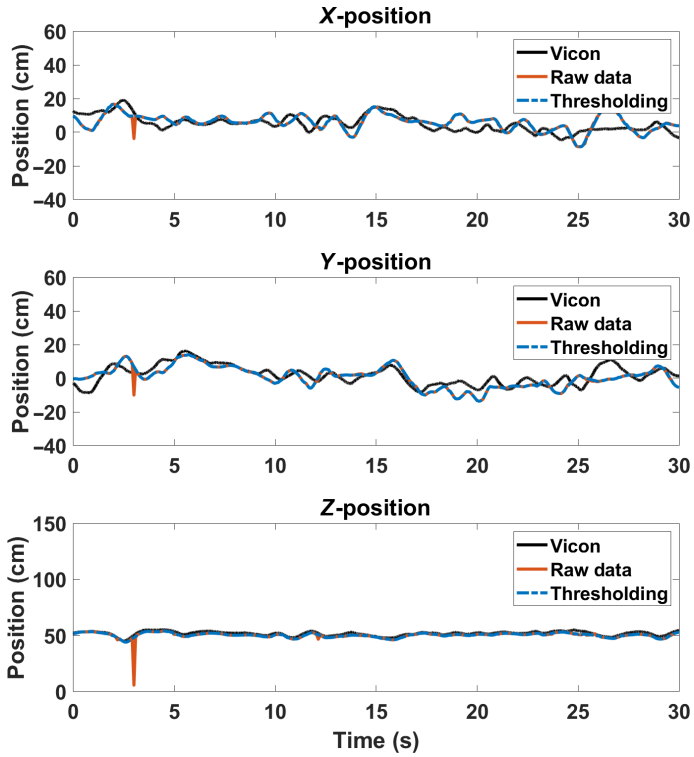


Fig. 9. Estimation of aircraft position relative to a stationary landing marker from onboard camera. Results are compared with and without the application of feature match and inlier thresholding.

Hand-Held Tests

A series of tests were conducted to evaluate the feature-based algorithm's performance in various scenarios in a controlled manner. Two cases were investigated and are discussed below: (1) tracking ship-deck motion in ideal visual conditions; (2) tracking under visually degraded conditions. The quadrotor was held by hand above the landing pad. This setup removes the effects of vibration and motion present in free flight, allowing evaluation of different scenarios without introducing additional difficulties. It also allows for exaggerated pitch and roll motions of the UAV to stress the robustness of the algorithm. Such motions would be impractical to recreate in free flight.

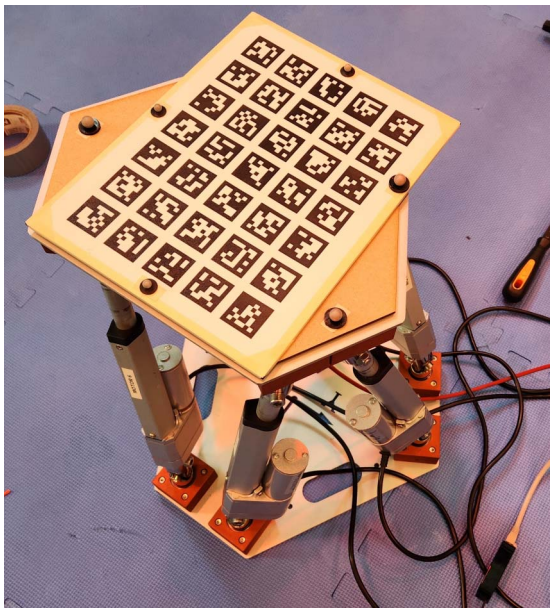


Fig. 10. Small Stewart platform with attached landing pad image.

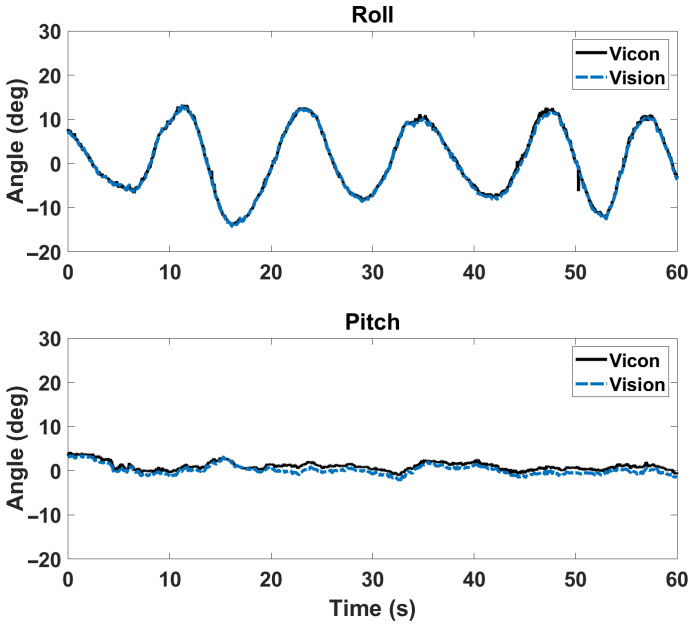


Fig. 11. Estimation of aircraft attitude angles relative to a moving Stewart platform from onboard camera, with aircraft held stationary.

For all tests, the landing pad image was attached to a small Stewart platform (Acrome Stewart Pro) capable of generating linear and rotational motions, shown in Fig. 10. The platform follows a motion trajectory representing Sea-state 6, obtained from ship-deck motion histories provided in the Navy SCONE database (Ref. 24). The motion is stochastic and predominantly features large roll motions of up to 15 deg with smaller variations in pitch and linear position. This allows algorithm performance to be evaluated with realistic ship-deck motions.

For clarity, only the vision algorithm output after match and inlier thresholding is shown in this section, labeled as "Vision." The Kalman filtered output is omitted.

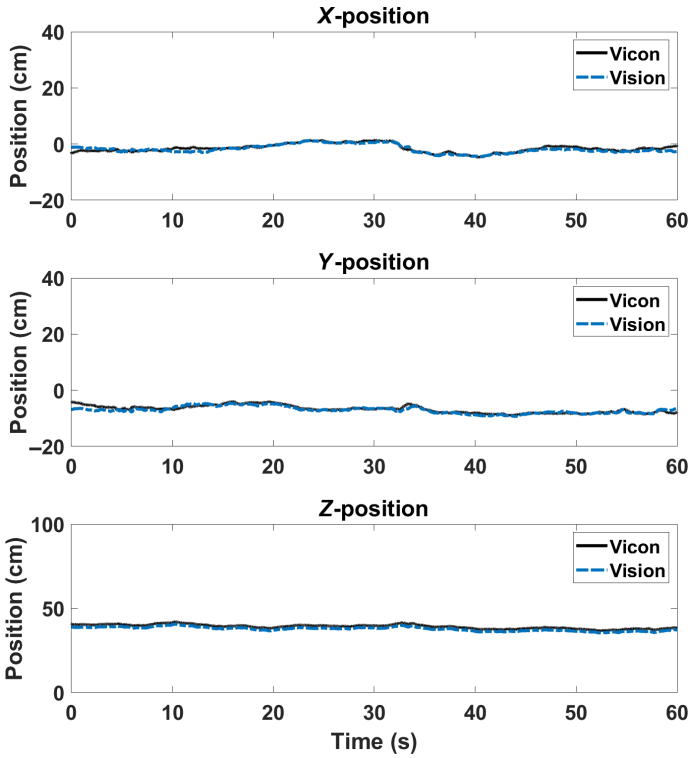


Fig. 12. Estimation of aircraft position relative to a moving Stewart platform from onboard camera, with aircraft held stationary.

Tracking ship-deck motion

After validating the baseline performance of the vision algorithm, it is then demonstrated for tracking ship-deck motion in ideal visual conditions. The Stewart platform used for these tests, with landing pad attached, is shown in Fig. 10.

First, the quadrotor is held stationary above the moving Stewart platform, simulating a stable hover above the ship-deck. In this scenario, the algorithm can correctly follow the movement of the platform, estimating both attitude and position with minimal error, as shown in Figs. 11 and 12.

Next, the quadrotor is moved with an arbitrarily varying pitch, roll, and linear motions above the stochastically moving platform. As the quadrotor was held by hand, it was possible to produce pitch motions of up to 15 deg and roll motions exceeding 20 deg. As shown in the results in Fig. 13, the algorithm is able to accurately follow complex variations of pitch and roll. The results for the Z-position are again very precise, as indicated in the results in Fig. 14. However, some errors appear in the estimates of X- and Y-positions.

The RMSE values, given in Table 3, provide a quantitative measure of algorithm performance in these scenarios. For ship-deck motion alone, the errors in pose estimation are very low. However, these errors increase with significant motion of both the ship-deck and quadrotor. During an actual landing, the quadrotor waits for a quiescent period when the deck's orientation and rate of change are small. As observed from the results, the algorithm is able to estimate orientation satisfactorily. In contrast, the estimate of linear position is used in the quadrotor control loop to track the deck, in order to remain above the target for landing. The position errors from the logged data are small compared to the quadrotor size; additionally, the quadrotor will land on a much larger deck than the one tested in these experiments. As a result, these errors are unlikely to interfere significantly with landing.

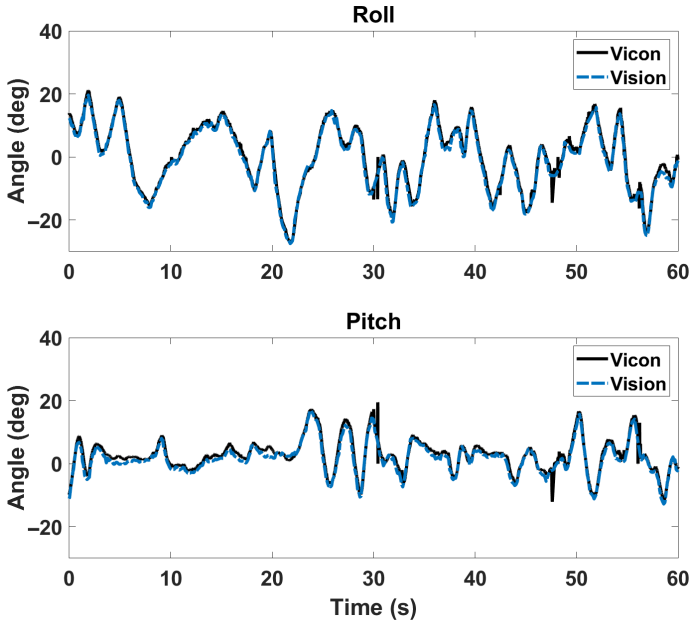


Fig. 13. Estimation of aircraft attitude angles relative to a moving Stewart platform from onboard camera, with aircraft also moving.

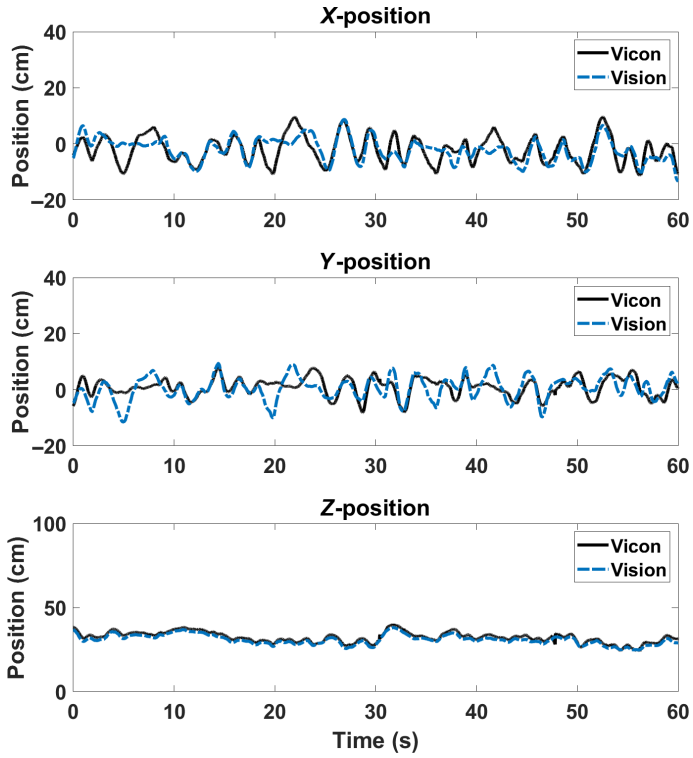


Fig. 14. Estimation of aircraft position relative to a moving Stewart platform from onboard camera, with aircraft also moving.

Table 3. RMSE for ship-deck motion

Test Case	*X* (cm)	*Y* (cm)	*Z* (cm)	Roll (deg)	Pitch (deg)	Yaw (deg)
Quadrotor stationary	0.6	0.5	1.4	0.5	1.0	1.3
Quadrotor moving	3.5	3.5	1.5	1.3	1.3	1.5

Table 4. RMSE for visually degraded test cases

Test Case	*X* (cm)	*Y* (cm)	*Z* (cm)	Roll (deg)	Pitch (deg)	Yaw (deg)
Low illumination	1.5	1.2	1.4	0.8	1.5	1.3
Glare	1.5	1.2	1.1	0.6	1.4	1.3
Occlusion (regular)	1.4	1.4	1.3	1.1	1.2	1.3
Occlusion (irregular)	1.5	1.9	1.7	1.7	2.0	1.7
Increased camera distance	1.4	1.2	1.2	1.9	1.9	1.4

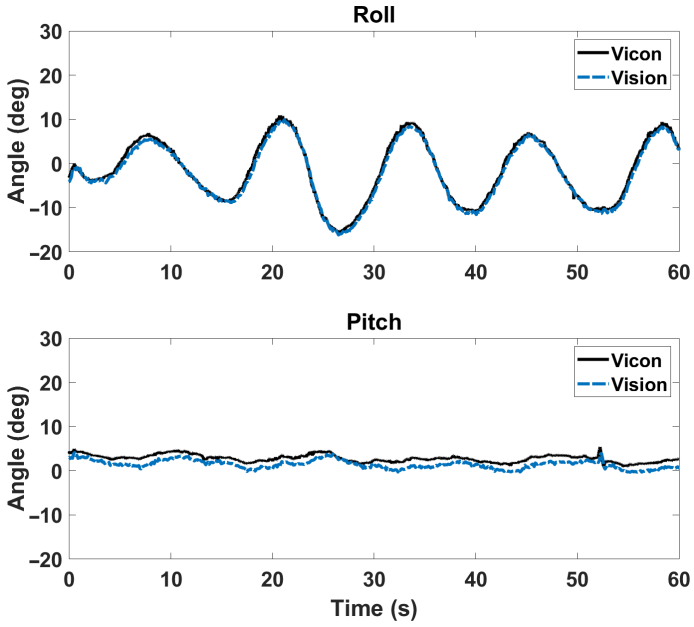


Fig. 15. Estimation of aircraft attitude angles relative to a landing pad undergoing ship-deck motion under low (10 lux) illumination.

Visually degraded conditions

A critical advantage of the feature-based algorithm is that it offers robustness to occlusion or varying lighting conditions, where the fiducial algorithm will fail outright. We previously demonstrated this robustness in benchtop tests. A series of tests were completed to analyze performance under visually challenging conditions onboard. In all tests, the quadrotor was held stationary above the moving Stewart platform. Three specific cases were evaluated: (1) low illumination, (2) glare, and (3) occlusion. The errors for each case are given in Table 4; each case is described in more detail below.

The first test evaluates the algorithm in low illumination conditions. Dark indoor lighting, approximately 10 lux, was used for this test case. The results of this test are provided in Figs. 15 and 16. The pose estimates are not significantly degraded by the low illumination condition, as shown by both the quantitative error calculations and the close match in the plots between the estimated and ground truth poses.

The second test evaluates the algorithm under glare. A 900-lumen flashlight was mounted above the landing pad, creating significant glare on the image, as shown in Fig. 17. This setup is intended to simulate glare, such as when bright lights or reflections are present on a real ship-deck. The test was also conducted under low illumination (10 lux) to increase the effect of this glare. The resulting attitude and position estimates are provided in Figs. 18 and 19. The estimates of attitude and

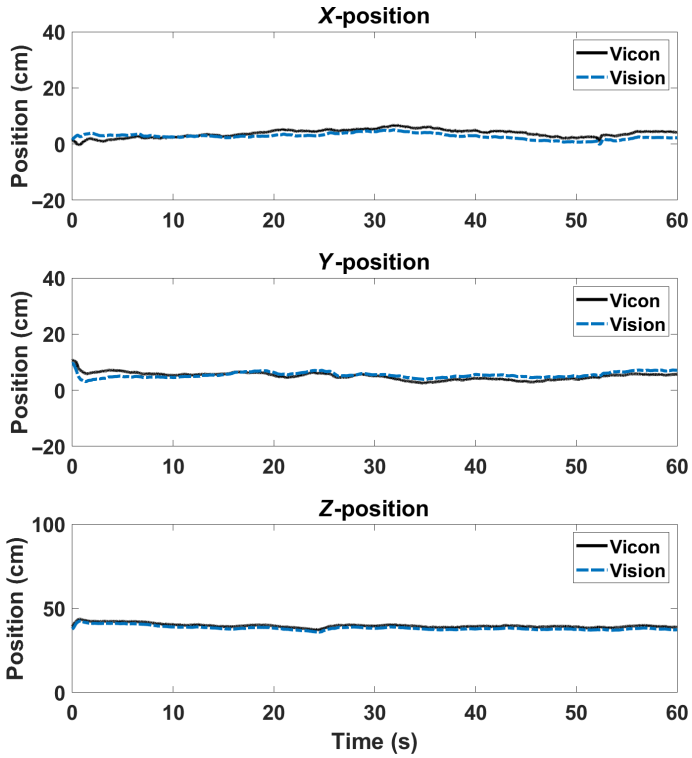


Fig. 16. Estimation of aircraft attitude angles relative to a landing pad undergoing ship-deck motion under low (10 lux) illumination.

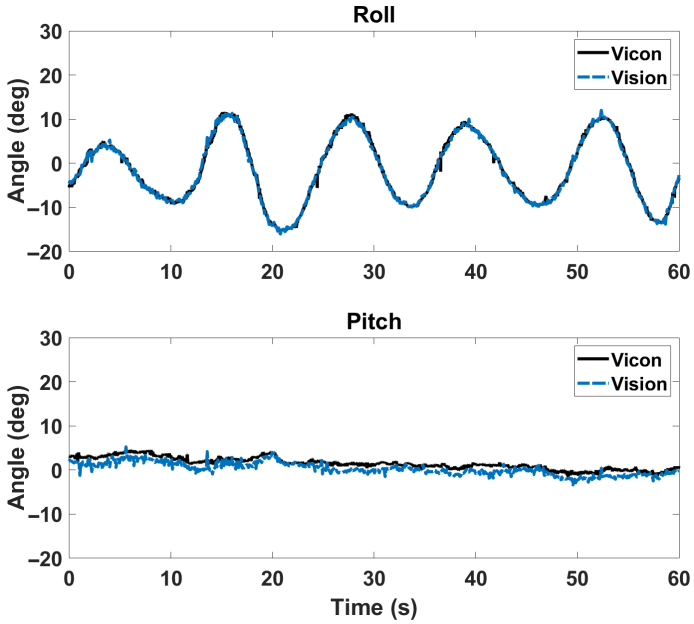


Fig. 18. Estimation of aircraft attitude angles relative to a landing pad undergoing ship-deck motion from onboard camera, with a 900-lumen light used to simulate glare.

position again closely follow the ground truth, indicating that the algorithm can perform acceptably in this condition. The RMSE results show comparable error to the low illumination case.

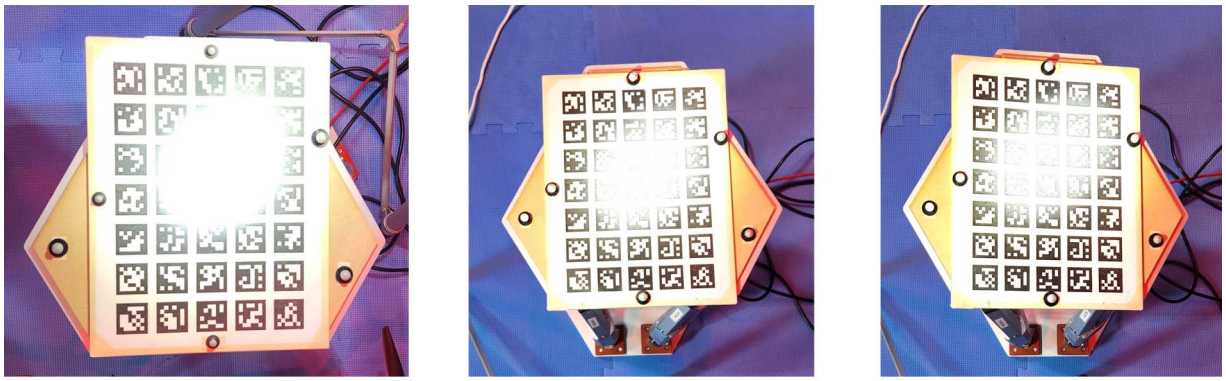
Finally, tests were completed to observe the performance of the algorithm for a landing pad under significant occlusion. Paper coverings are placed to occlude a portion of the image, as shown in Fig. 20. Two test cases were evaluated: (1) regular occlusion and (2) irregular occlusion. The tests were conducted in normal indoor lighting (300 lux) to avoid confusing the effects of occlusion and low illumination.

For the first case, shown in Fig. 20(a), 20 of the 35 tags (57% of the image) are uniformly covered. In the second case shown in Fig. 20(b), a number of shapes are arranged in a diagonal pattern, partially or fully occluding all of the tags on the image. These shapes cover approximately 49% of the image. Note that, due to the partial obstruction of all tags, a fiducial algorithm will fail outright under this condition.

The results of the first case are shown in Figs. 21 and 22. As in the previous two cases, the results accurately follow the ground truth data, and the RMSE suggests similar levels of error to the previous cases. The results for the second case are given in Figs. 23 and 24. The estimates of roll and pitch are noisier than in the previous case, and, while still accurate overall, display increased error compared to ground truth. The RMSE results also show increased error for both attitude and position estimates compared to the previous case.

Increasing camera distance

For the previous hand-held tests, the camera was held at a vertical distance of approximately 40 cm from the landing pad. This condition simulates the terminal landing phase, in which the aircraft is hovering



(a) (b) (c)

Fig. 17. (a) Landing pad image with significant glare created by a 900-lumen light. (b) and (c) Change in glare pattern due to platform motion during a typical test.

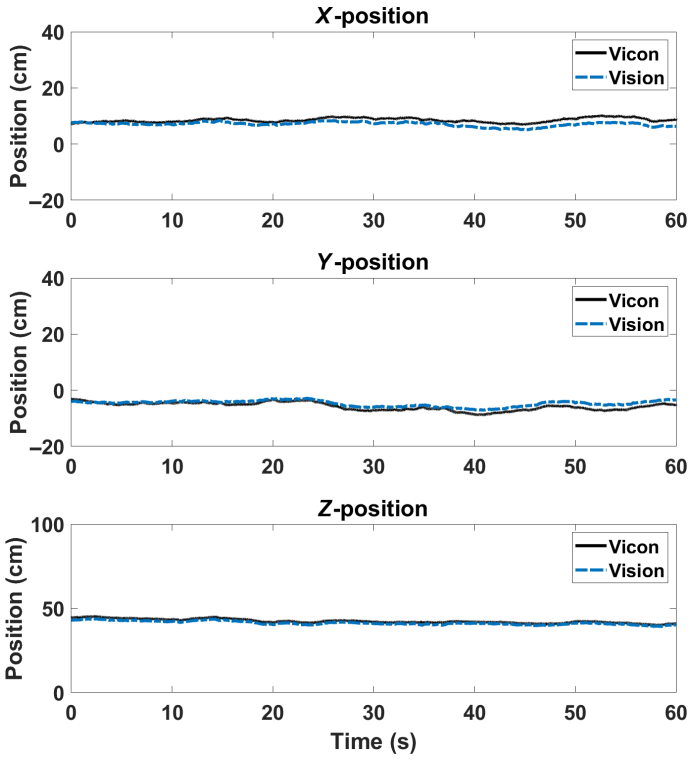


Fig. 19. Estimation of aircraft position relative to a landing pad undergoing ship-deck motion from onboard camera, with a 900-lumen light used to simulate glare.

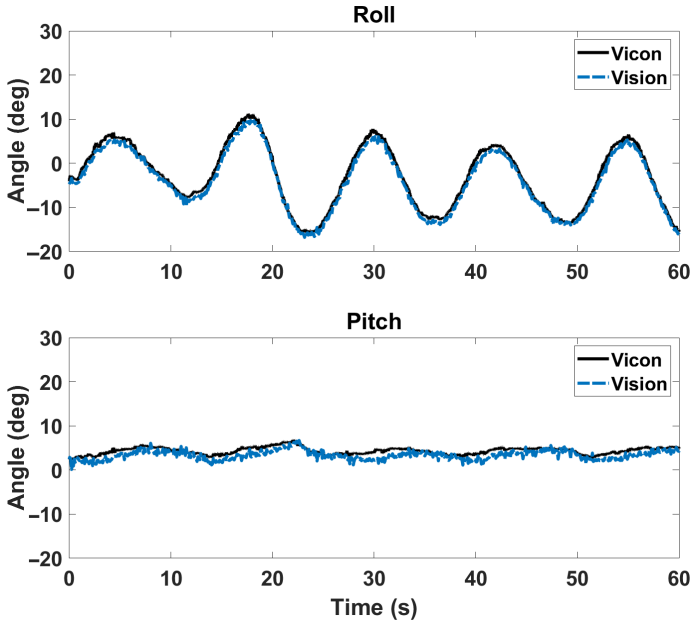


Fig. 21. Estimation of aircraft attitude angles relative to an occluded landing pad undergoing ship-deck motion. The occlusions uniformly cover 57% of the marker patterns on the landing pad image.

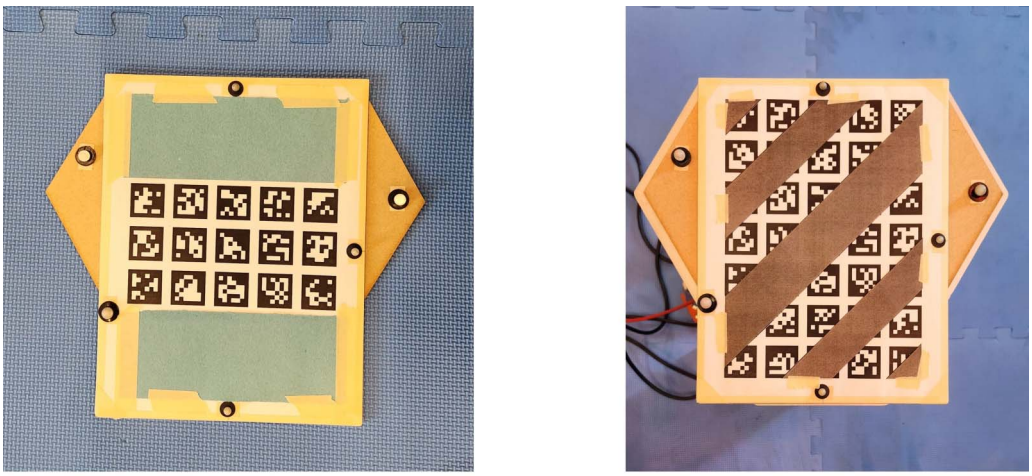
closely above the ship-deck. However, the effect of increasing this distance is also of interest, as the feature-matching approach may also be used during earlier landing phases in which the camera is further from the landing pad. Therefore, a test was completed to demonstrate the effect of increasing the camera distance from the landing pad image. For this test, the vertical (Z-axis) distance was increased to approximately 85 cm, double the distance for the previous tests.

The results of this test are provided in Figs. 25 and 26. The attitude estimation results are accurate overall but show increased noise compared to previous tests. The position results follow the ground truth closely. The RMSE results further confirm these observations; the position estimation error remains low, but the error in attitude estimation is similar to that of the irregular occlusion case.

Feature match and inlier summary

As the count of feature matches and inliers has a significant effect on the quality of pose estimates, analyzing this data for different test cases can provide additional insight. Mean counts of feature matches and inliers are provided in Table 5.

From this data, a general correlation can be observed between attitude estimation error and feature match and inlier counts. In cases with low numbers of feature matches, specifically the irregular occlusion and



(a) (b)

Fig. 20. Occluded landing pad images used to evaluate algorithm performance. (a) Occlusions uniformly covering 20 of the 35 tags, or 57% of the landing pad image. (b) Irregular occlusions partially or fully covering all tags. These occlusions cover 49% of the image.

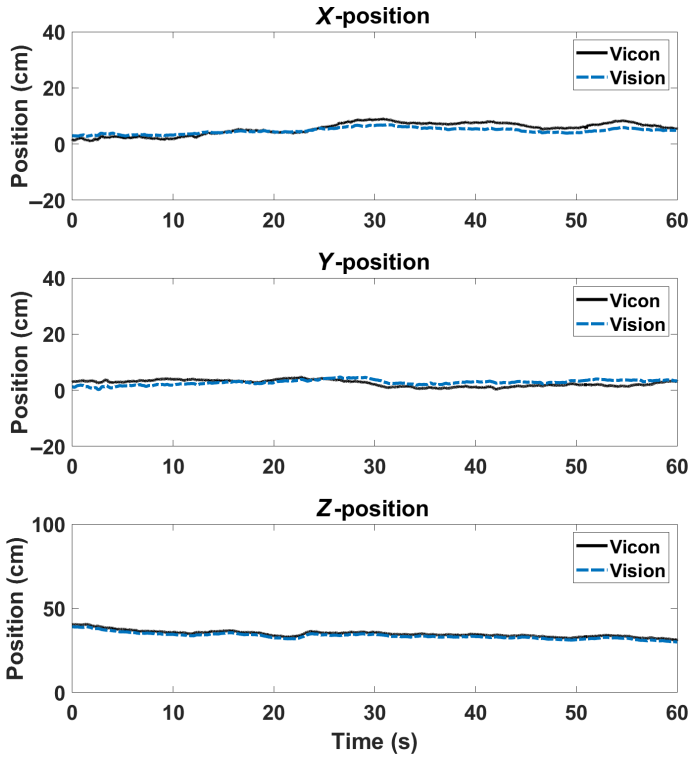


Fig. 22. Estimation of aircraft position relative to an occluded landing pad undergoing ship-deck motion. The occlusions uniformly cover 57% of the marker patterns on the landing pad image.

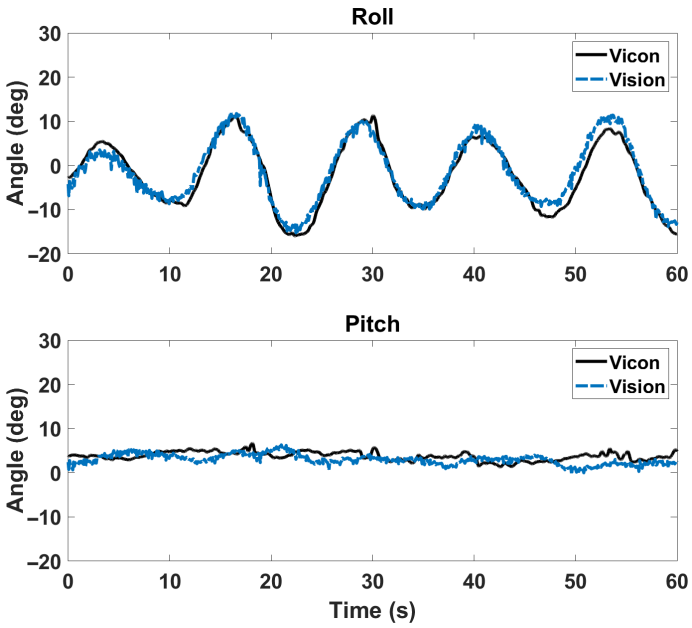


Fig. 23. Estimation of aircraft attitude angles relative to an occluded landing pad undergoing ship-deck motion. Irregular occlusions partially or fully cover all marker patterns on the landing pad image.

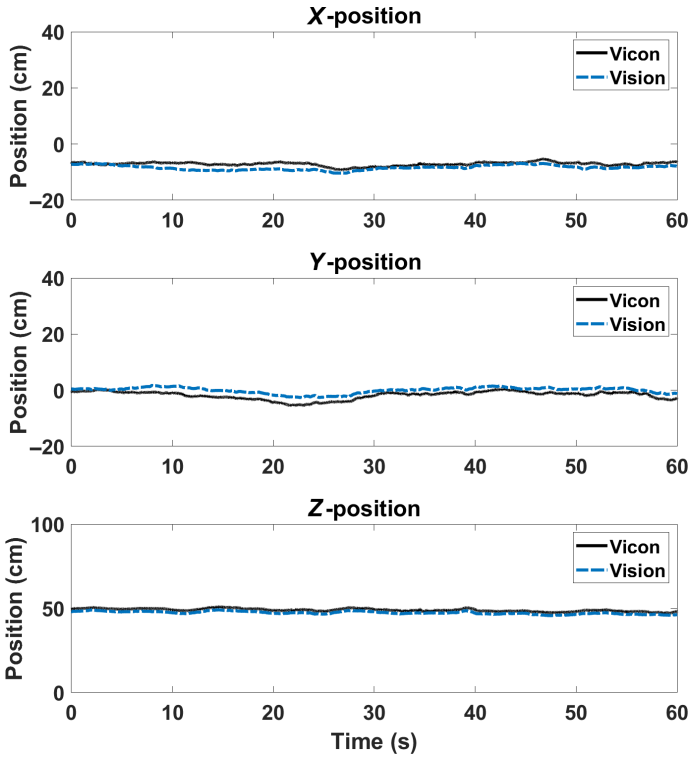


Fig. 24. Estimation of aircraft position relative to an occluded landing pad undergoing ship-deck motion. Irregular occlusions partially or fully cover all marker patterns on the landing pad image.

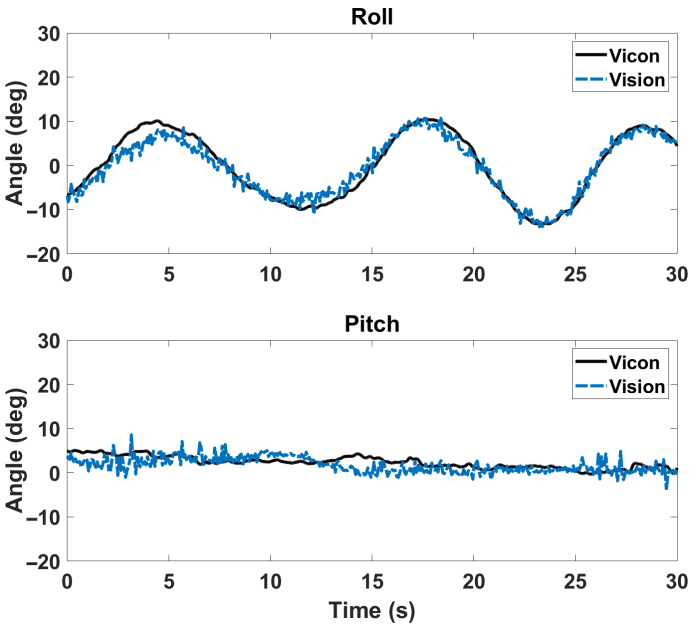


Fig. 25. Estimation of aircraft attitude angles relative to a landing pad undergoing ship-deck motion, with the *Z*-axis (vertical) distance increased from previous tests.

increased distance cases, the RMSE for roll and pitch is notably larger than the other cases.

However, a similar correlation is not apparent for position estimates. The RMSE for linear position is comparable for all visually degraded test cases, excluding the irregular occlusion case. This is despite the significant variation in mean number of feature matches. The error in linear position is much greater for the quadrotor moving case. For this case, the large-angle motions introduced factors such as motion blur, large out-of-plane rotations, and the landing image partially moving outside the camera field of view. This indicates that these conditions have a

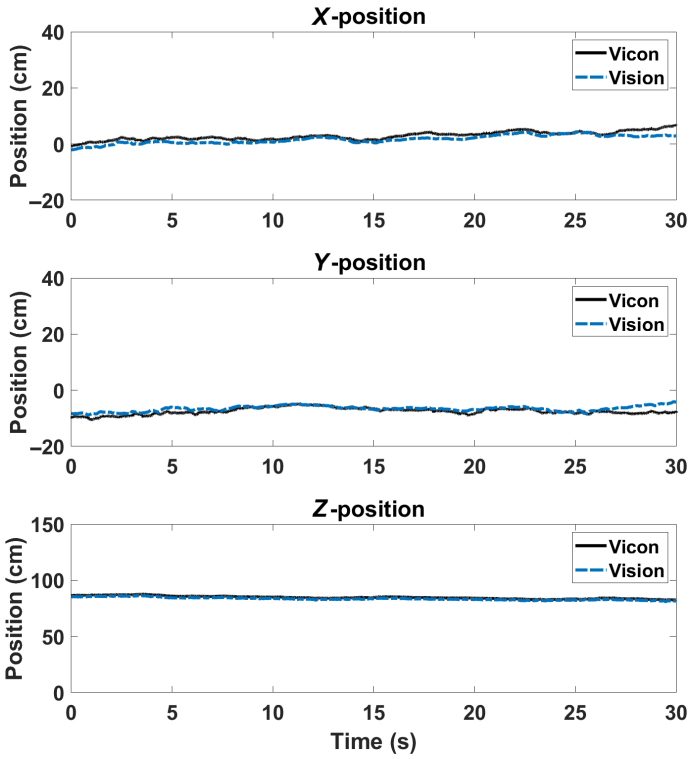


Fig. 26. Estimation of aircraft position relative to a landing pad undergoing ship-deck motion, with *Z*-axis (vertical) distance increased from previous tests.

Table 5. Summary of mean feature match and inlier counts for different test cases

Test Case	Mean Feature Matches	Mean Inliers
Quadrotor stationary	346	302
Quadrotor moving	252	197
Low illumination	360	317
Glare	154	127
Occlusion (regular)	199	161
Occlusion (irregular)	65	50
Increased camera distance	44	37

major effect on the accuracy of position estimates. The error also increases modestly in the irregular occlusion case. This suggests that occlusions interfering with larger patterns on the landing image, and by extension larger features, may have a moderate effect on position estimation error.

Free-Flight Hover Tests

The algorithm was next validated during hovering flight. The quadrotor was manually piloted through takeoff, hover above the image, and landing. Algorithm output was logged during the entire flight for each test. Only the hover segment is presented here since the goal of these experiments is to evaluate performance in hover.

Baseline conditions

Results were initially collected for ideal visual conditions, indoor lighting (300 lux), and no occlusion of the pad. For this test, the landing

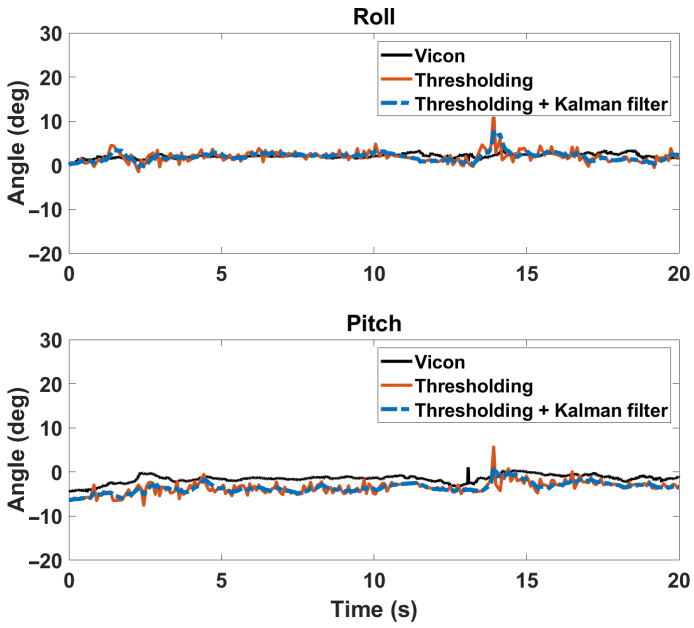


Fig. 27. Estimation of hovering aircraft attitude angles relative to a stationary landing marker in indoor illumination (300 lux).

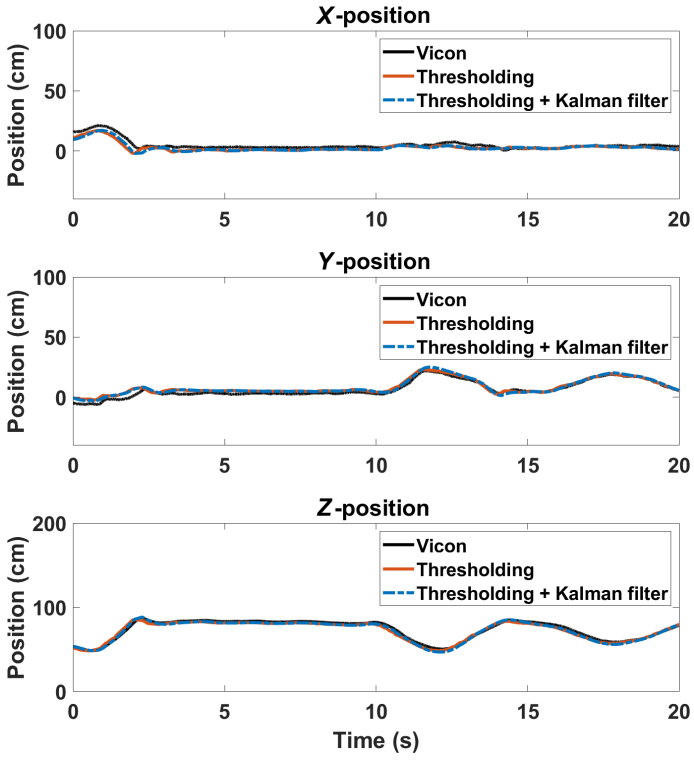


Fig. 28. Estimation of hovering aircraft position relative to a stationary landing marker in indoor illumination (300 lux).

image was secured to the ground so no deck motions were present. The results of this test are shown in Figs. 27 and 28. The estimated pose, both attitude and position, closely follows the ground truth. Quantitatively, this data shows an RMSE of 2 cm in each linear axis and an RMSE of 2.3 deg and 1.3 deg in pitch and roll, respectively. While some minor noise is present in the raw estimates for pitch and roll due to quadrotor

Table 6. RMSE for free flight in visually degraded conditions

Test Case	*X* (cm)	*Y* (cm)	*Z* (cm)	Roll (deg)	Pitch (deg)
Low illumination only	3.0	3.0	1.7	2.0	1.2
Occlusion only	0.9	1.0	2.1	1.1	1.2
Low illumination and occlusion	1.3	1.3	3.0	2.2	2.0

vibration as well as the increase in *Z*-position compared to the hand-held test condition, this noise is mostly mitigated by the Kalman filter. This indicates that, under ideal visual conditions, vibration and small motions from actual quadrotor hover have a minimal effect on the quality of algorithm output.

Visually degraded conditions

Next, visually degraded conditions were introduced. Occlusion and reduced illumination were considered, as these conditions were the most challenging in the controlled hand-held test cases earlier. As in the previous test, the landing pad was fixed to the ground. The quantitative RMSE results for all cases are given in Table 6.

A test was first completed in dark indoor illumination (10 lux) for a nonoccluded landing pad. These results are shown in Figs. 29 and 30. Additional tests were completed for an occluded landing pad in indoor illumination (300 lux), shown in Figs. 31 and 32, and dark indoor illumination (10 lux), shown in Figs. 33 and 34. The landing pad was occluded by covering 57% of the tags on the image.

For all cases, the estimated position closely follows the ground truth position, showing no significant change in performance with visually degraded conditions.

For occlusion in indoor lighting, some noise is present in the raw estimates of roll and pitch, though this noise is mostly mitigated by the Kalman filter. With a combination of low illumination and occlusion, an RMSE of 2.2 deg in pitch and 2.0 deg in roll is observed, higher than with occlusion or low illumination alone. However, some individual attitude estimates show errors exceeding 10 deg. Errors such as this could cause the controller to falsely reject a period of calm deck motion for landing. While this in itself will not result in a crash, this could make it more difficult to autonomously carry out a landing.

Compared to the hand-held tests, the quality of pose estimation decreases somewhat in free flight. Quantitatively comparing the RMSE results, while the pose estimation errors are still low overall, the errors do increase due to vibrations and aircraft motions in free flight.

Ship-deck motion

Finally, a test was completed to evaluate algorithm performance in hover for ship-deck motion. For this test, the landing pad was attached to a Stewart platform following the same Sea-state 6 motion history used for the hand-held tests. The test was conducted under ideal visual conditions, indoor lighting (300 lux), and with no occlusion of the landing image. The results of this test are given in Figs. 35 and 36. As in the previous tests, the results follow the ground truth accurately. For the position estimates, this data shows an RMSE of 3.1 cm in the *X*-direction, 2.9 cm in the *Y*-direction, and 3.7 cm in the *Z*-direction. For attitude estimates, an RMSE of 1.8 deg in pitch and 1.9 deg in roll is observed. These errors show only a modest increase from the baseline case. The feature-based algorithm is, therefore, capable of accurately estimating the pose of a moving ship-deck in free hover.

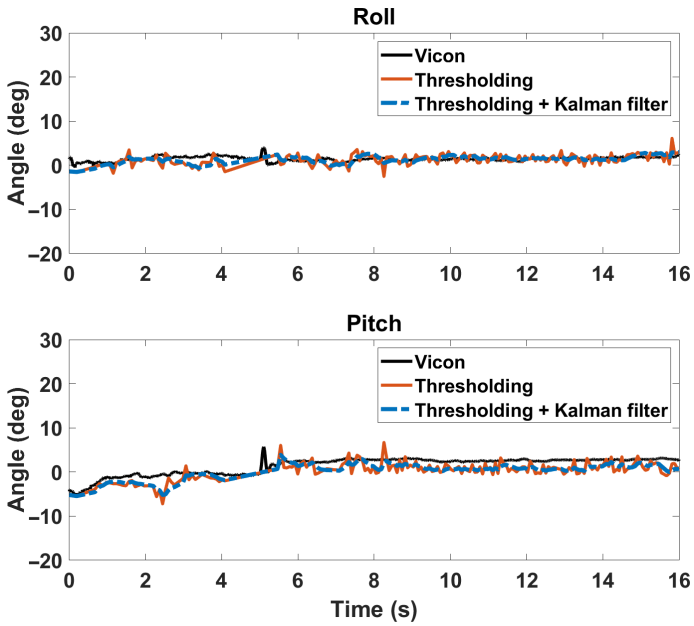


Fig. 29. Estimation of hovering aircraft attitude angles relative to a stationary landing marker in low illumination (10 lux).

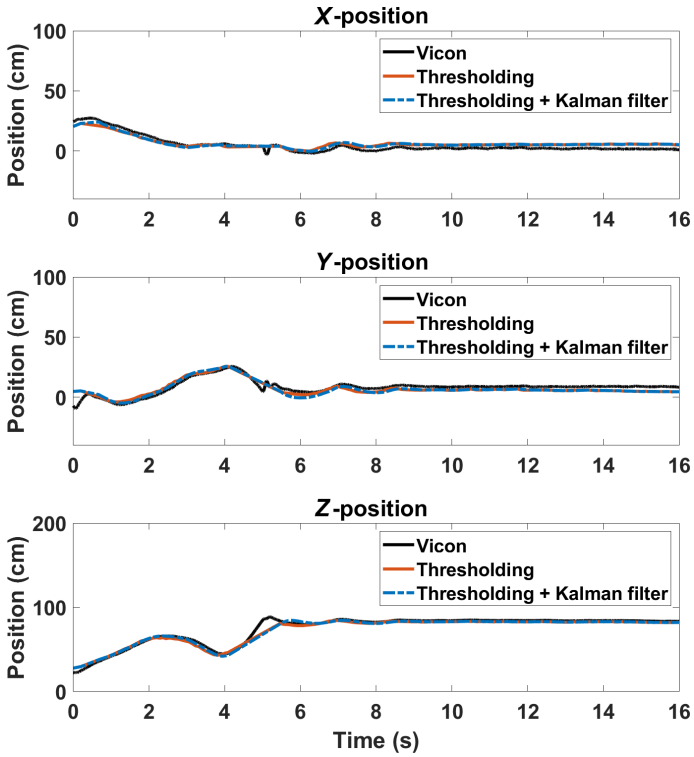


Fig. 30. Estimation of hovering aircraft position relative to a stationary landing marker in low illumination (10 lux).

Performance Comparison with Fiducial-Based Approaches

Previous work on vision-based landing using fiducial markers can provide a useful benchmark for the performance of the feature-based algorithm. The feature-based algorithm offers advantages over fiducial-based techniques, namely robustness to visual degradation; additionally, the chosen landing pad image resembles typical fiducial marker designs. Therefore, it is important to investigate if the feature-based

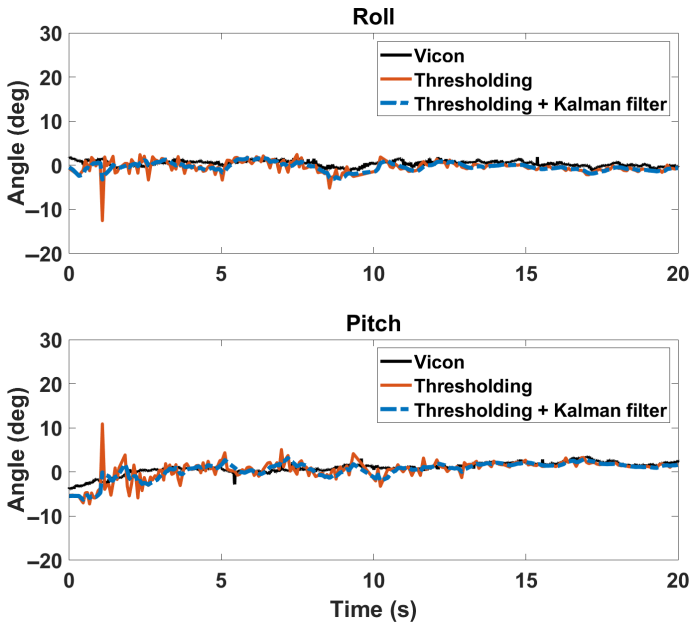


Fig. 31. Estimation of hovering aircraft attitude angles relative to a stationary, 57% occluded landing marker in indoor illumination (300 lux).

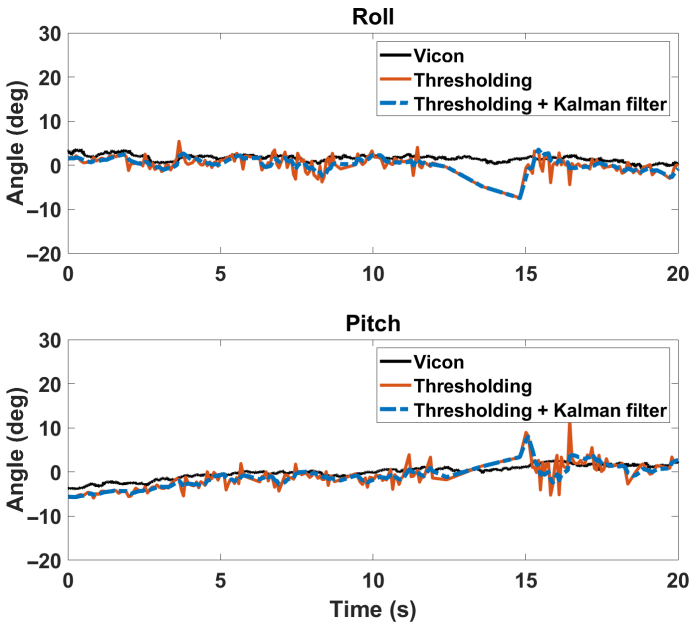


Fig. 33. Estimation of hovering aircraft attitude angles relative to a stationary, 57% occluded landing marker in low illumination (10 lux).

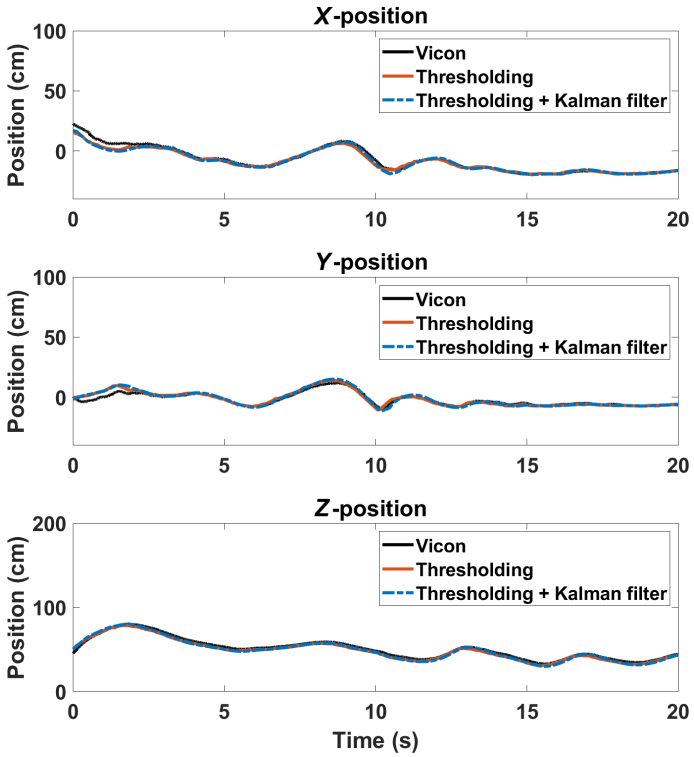


Fig. 32. Estimation of hovering aircraft position relative to a stationary, 57% occluded landing marker in indoor illumination (300 lux).

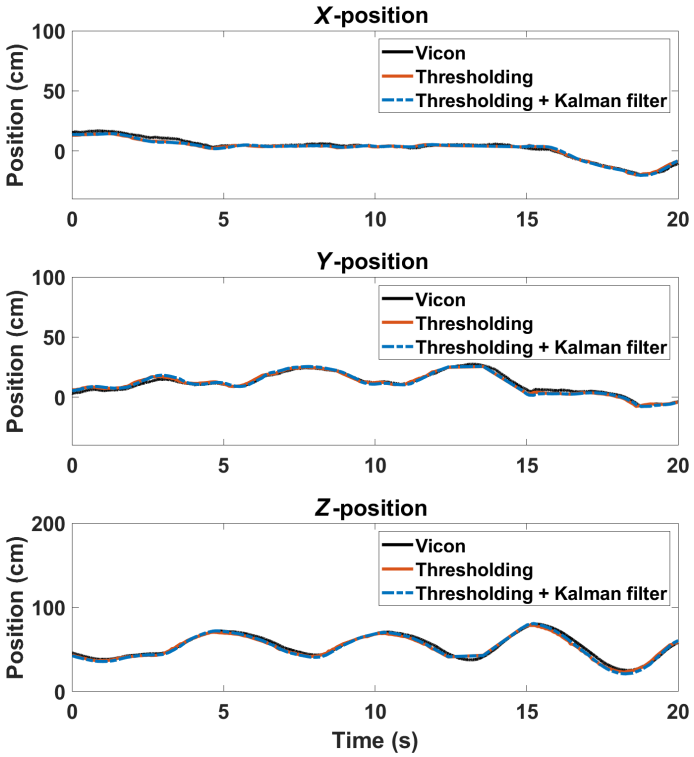


Fig. 34. Estimation of hovering aircraft position relative to a stationary, 57% occluded landing marker in low illumination (10 lux).

technique offers comparable pose estimation accuracy to fiducial-based approaches.

Our previous fiducial-based algorithm, evaluated under oscillating motion in visually ideal conditions, produced pose estimates with error of approximately 1–2 cm along each linear axis, 1–2 deg in pitch, and 5 deg in roll (Ref. 18). Araar et al. utilized a landing pattern consisting of multiple AprilTag markers for landing on a linearly moving (non-rotating) platform, reporting a pose estimation error of approximately 1 cm for each linear axis and 0.5–7 deg for rotation (Ref. 1). Nicholson et al. estimated the pose of a ship-deck at varying sea-states using a

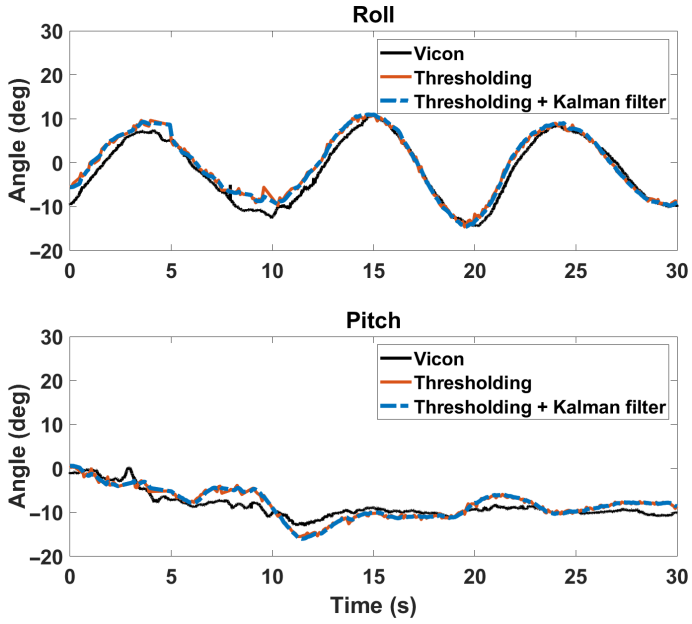


Fig. 35. Estimation of hovering aircraft attitude angles relative to a landing pad undergoing ship-deck motion.

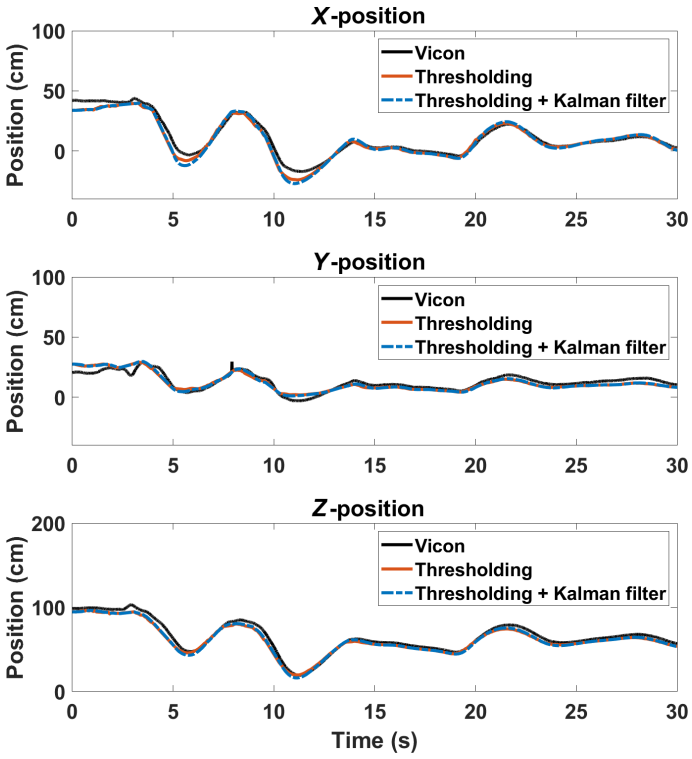


Fig. 36. Estimation of hovering aircraft position relative to a landing pad undergoing ship-deck motion.

recursive AprilTag fiducial marker array; vision-based estimates were fused with inertial data for improved estimation quality. The results found an average error of 8–23 cm for position estimates and 1.8–3.1 deg for orientation (Ref. 10).

For the feature-based algorithm, error across various test scenarios was found to be less than 4 cm along each linear axis and 2.5 deg for pitch and roll. Based on these results, the feature-based algorithm demonstrates comparable pose estimation accuracy to approaches using fiducial markers.

Conclusions

This work refined and extensively characterized a 2D feature-based vision algorithm for ship-deck landing under degraded visual conditions. Systematic tests were carried out on a quadrotor uncrewed aerial system developed in-house which incorporates a computationally powerful flight computer. All computations were onboard. A series of real-time tests, with the quadrotor moved by hand for large angles, were completed to first evaluate the algorithm's performance. Free-flight tests were then carried out. From these results, the following key conclusions are drawn:

1) A feature-based algorithm is capable of accurately estimating the pose of a stochastically moving platform at Sea-state 6 in both controlled and free-flight conditions.

2) A reference image with many distinct, high-contrast features is ideal for good algorithm performance. While the algorithm can function with typical ship-deck markings, performance degrades as these patterns have few good features.

3) In ideal visual conditions, the vision algorithm could successfully estimate large variations in attitude in excess of 20 deg.

4) The performance of the algorithm is resilient to visually challenging conditions. In hand-held tests, where roll motions of up to 15 deg were present, the ship-deck pose could be accurately estimated with low illumination, glare, and occlusion of the landing pad as well as with increased camera distance. The deck pose could also be accurately estimated in free-flight hover with occlusion and low illumination.

5) Vibrations and small motions introduced in free hover have a minor effect on algorithm performance, Noise in algorithm output can be mitigated through Kalman filtering.

6) A time delay of approximately 400 ms was present in the vision algorithm output. This was caused by a combination of camera latency and algorithm computation time. However, this delay is not expected to significantly affect tracking and landing.

7) With onboard computer and vision hardware, the algorithm updates at a speed of 9–11 Hz, which is adequate for capturing ship-deck motion.

8) The pose estimation results can be significantly improved by incorporating thresholds for feature matches and inliers; such thresholding allows poor pose estimates to be automatically rejected.

Acknowledgments

This work was supported under the NAVAIR Grant N0042121S0001 with technical monitoring from James Bumbaugh, Daniel Shafer, and Peter Arslanian, and partly under the Army/Navy/NASA Vertical Lift Research Center of Excellence (VLRCOE) number W911W61120012 with technical monitoring from Dr. Mahendra Bhagwat.

References

[1]Araar, O., Aouf, N., and Vitanov, I., "Vision Based Autonomous Landing of Multirotor UAV on Moving Platform," *Journal of Intelligent and Robotic Systems*, Vol. 85, February 2017, pp. 369–384, DOI: 10.1007/s10846-016-0399-z.

[2]Badakis, G., Koutsoubelias, M., and Lalis, S., "Robust Precision Landing for Autonomous Drones Combining Vision-based and Infrared Sensors," Proceedings of the 2021 IEEE Sensors Applications Symposium, Sundsvall, Sweden, August 23–25, 2021.

[3]Lin, J., Wang, Y., Miao, Z., Zhong, H., and Fierro, R., "Low-Complexity Control for Vision-Based Landing of Quadrotor UAV on Unknown Moving Platform," *IEEE Transactions on Industrial Informatics*, Vol. 18, (8), August 2022, pp. 5348–5358, DOI: 10.1109/TII.2021.3129486.

[4]Lee, H., Jung, S., and Shim, D. H., "Vision-based UAV Landing on the Moving Vehicle," Proceedings of the 2016 International Conference on Unmanned Aircraft Systems, Arlington, VA, June 7–10, 2016.

[5]Falanga, D., Zanchettin, A., Simovic, A., Delmerico, J., and Scaramuzza, D., "Vision-Based Autonomous Quadrotor Landing on a Moving Platform," Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics. Shanghai, China, October 11–13, 2017.

[6]Kawamura, E., Kannan, K., Lombaerts, T., and Ippolito, C., "Vision-Based Precision Approach and Landing for Advanced Air Mobility," AIAA 2022-0497, Proceedings of the AIAA SciTech 2022 Forum, San Diego, California, January 3–7, 2022.

[7]Cesetti, A., Frontoni, E., Mancini, A., Zingaretti, P., and Longhi, S., "A Vision-Based Guidance System for UAV Navigation and Safe Landing Using Natural Landmarks," *Journal of Intelligent and Robotic Systems*, Vol. 57, January 2010, pp. 233–257, DOI: 10.1007/s10846-009-9373-3.

[8]Truong, N. Q., Nguyen, P. H., Nam, S. H., and Park, K. R., "Deep Learning-Based Super-Resolution Reconstruction and Marker Detection for Drone Landing," *IEEE Access*, Vol. 7, May 2019, pp. 61639–61655. DOI: 10.1109/ACCESS.2019.2915944.

[9]Cho, G., Choi, J., Bae, G., and Oh, H., "Autonomous Ship Deck Landing of a Quadrotor UAV Using Feed-Forward Image-Based Visual Servoing," *Aerospace Science and Technology*, **130** 107869 (2022), DOI: 10.1016/j.ast.2022.107869.

[10]Nicholson, D. J., Hendrick, C. M., Jaques, E. R., Horn, J. F., and Langelaan, J. W., "Scaled Experiments in Vision-Based Approach and Landing in High Sea States," AIAA 2022-3280, Proceeding of the AIAA AVIATION 2022 Forum, Chicago, IL, June 27–July 1, 2022.

[11]Sanchez-Lopez, J., Pestana, J., Saripalli, S., and Campoy, P., "Toward Visual Autonomous Ship Board Landing of a VTOL UAV," *Journal of Intelligent and Robotic Systems*, Vol. 74, April 2014, pp. 113–127, DOI: 10.1007/s10846-013-9926-3.

[12]Lin, S., Garratt, M. A., and Lambert, A. J., "Monocular Vision-Based Real-Time Target Recognition and Tracking for Autonomously Landing an UAV in a Cluttered Shipboard Environment," *Autonomous Robots*, Vol. 41, April 2016, pp. 881–901, DOI: 10.1007/s10514-016-9564-2.

[13]Meng, Y., Wang, W., Han, H., and Ban, J., "A Visual/Inertial Integrated Landing Guidance Method for UAV Landing on the Ship," *Aerospace Science and Technology*, Vol. 85, January 2019, pp. 474–480, DOI: 10.1016/j.ast.2018.12.030.

[14]Holmes, W. K., and Langelaan, J. W., "Autonomous Ship-board Landing using Monocular Vision," Proceedings of the 72nd Annual Forum of the American Helicopter Society, West Palm Beach, FL, 17–19, 2016.

[15]Truong, Q., Rakotomomonjy, T., Taghizad, A., and Biannic, J.-M. "Vision-Based Control for Helicopter Ship Landing with Handling Qualities Constraints," *IFAC-PapersOnLine*, Vol. 49, (17), December 2016, pp. 118–123, DOI: 10.1016/j.facol.2016.09.021.

[16]Lee, B., Saj, V., Kalathil, D., and Benedict, M., "Intelligent Vision-Based Autonomous Ship Landing of VTOL UAVs," *Journal of the American Helicopter Society*, **68**, 022010 (2023), DOI: 10.4050/JAHS.68.022010.

[17]Wickramasuriya M., Taeyoung, L., and Snyder, M., "Deep Monocular Relative 6D Pose Estimation for Ship-Based Autonomous UAV," AIAA 2024-2877, Proceedings of the AIAA SciTech 2024 Forum, Orlando, FL, January 8–12, 2024.

[18]Shastry, A. K., Datta, A., and Chopra, I., "Vision-Based Autonomous UAS Landing on a Stochastically Moving Platform," Proceedings of the 77th Annual Forum of the Vertical Flight Society, Palm Beach, FL, May 2021.

[19]Shastry, A. K., Datta, A., and Chopra, I., "Feature-Based Vision For Stochastic Motion Tracking Under Partial Occlusion", Proceedings of the 78th Annual Forum of the Vertical Flight Society, Ft. Worth, TX, May 10–12, 2022.

[20]Britcher, V., Shastry, A. K., and Chopra, I., "Exploration of Feature-Based Algorithm for Autonomous Ship-Deck Landing under Visually Degraded Conditions," Proceedings of the 79th Annual Forum of the Vertical Flight Society, West Palm Beach, FL, May 16–18, 2023.

[21]Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, (2), November 2004, pp. 91–110, DOI: 10.1023/B:VISI.0000029664.99615.94.

[22]Fischler, M. A, and Bolles, R. C., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, (6), June 1981, pp. 381–395, DOI: 10.1145/358669.358692.

[23]Collins, T., and Bartoli, A., "Infinitesimal Plane-Based Pose Estimation," *International Journal of Computer Vision*, Vol. 109, (3), September 2014, pp. 252–286, DOI: 10.1007/s11263-014-0725-5.

[24]Schwartz, A., "SCONE - Standard Deck Motion Data for a Generic Surface Combatant," Office of Naval Research, March 2017.