GNSS-Independent Absolute Localization for Autonomous Quadcopters via Satellite Image Registration

Prepared by:

Pavle Slepcev / Marcus Wahila

University of Pittsburgh

Swanson School of Engineering

June 2026

Contents

1 Abstract

Conventional unmanned aerial vehicles (UAVs) depend on Global Navigation Satellite Systems (GNSS) and radio links for navigation. Both degrade in dense urban environments, in enclosed spaces, under heavy precipitation, and behind masking terrain, and both are vulnerable to jamming, spoofing, and electromagnetic interference. This has led to the development of alternatives such as frequency hopping, directional antennas, and fiber-optic tethers, but they introduce limitations of their own. Relative methods, such as visual odometry, avoid these links but accumulate unbounded drift over a flight. This report presents a low-cost quadcopter navigation system implementation that computes an absolute position fix from each camera frame by registering a live image against pre-downloaded, georeferenced satellite imagery, so error does not compound and the accuracy is set by the registration itself. Because the camera and satellite views observe the same ground plane, the two images are related by a homography estimated from ORB feature correspondences filtered by Lowe's ratio test and RANSAC geometric verification. We implemented this approach on a prototype built by modifying a DJI Phantom 3 to carry an NVIDIA Jetson Orin Nano companion computer and an Arducam IMX477 camera on a carbon-fiber-reinforced nylon mount, designed through incremental dead-load testing to a 560-gram payload limit and tuned to an 18 Hz isolation frequency to suppress motor and blade-pass vibration. A ROS 2 pipeline performs onboard image correction, contrast normalization, satellite-tile mosaicking, feature matching, and planar pose recovery in real time. Test flights demonstrated live camera-to-satellite matching with an average 8-meter error from DJI GPS used as ground truth, establishing the feasibility of drift-free, GNSS-independent localization on commodity hardware.

2 Introduction

Technology has come a long way in introducing humans to flight. We are here to talk about a different flying machine, a UAV. The first unmanned aerial vehicle(UAV) was invented in 1917 during World War I. It was a radio-controlled aircraft meant to be a pilotless bomb. Since then, drones have entered a new realm of design and importance. Today, drones are incredibly important in everything from agriculture and landscape mapping to search-and-rescue missions and military operations. As UAV technology continues to expand, their influence on the world will as well.

To fully describe this project, we have to dive into how drones work and what the modern-day problem in their structure is. Most average drones work through a GCS(Ground control system) or satellite link, which works through a radio connection between the controller and the drone or a low Earth orbit satellite. Positioning itself normally comes from a Global Navigation Satellite System (GNSS), the family of satellite constellations that includes GPS. Those signals are extremely weak by the time they reach the ground, so they diminish in dense urban environments, in enclosed spaces, under heavy precipitation, and behind masking terrain. Beyond these natural limits, the reliance on both the radio link and the GNSS signal can be attacked directly: jamming drowns them out, and spoofing replaces them with false data.

There have been alternatives to help prevent the effects of these obstructions, such as frequency hopping, directional antennas, and fiber-optic cables. If these countermeasures already exist, it is reasonable to ask what a new approach adds. Well, each one carries its own problems; for the use of fiber-optic to bypass electronic jamming, these are major mechanical limits in cable length, cables getting stuck during flight, operator exposure risks, and long-term environmental pollution. Using frequency hopping brings heavy hardware demands, synchronization vulnerabilities, and limited effectiveness against advanced wide-band or intelligent jamming tactics in dynamic electronic environments. Finally, using directional antennas creates issues such as coverage blind spots, orientation sensitivity, unpredictable reflection issues, and physical mounting constraints. The approach described here avoids these constraints at the cost of others(discussed in Section 6).

A different way to avoid these links is odometry, which estimates motion incrementally: it measures how far the vehicle has moved between successive observations and sums those increments to track position. Visual odometry and inertial dead reckoning both work this way. They require no external signal and are therefore immune to jamming, but because every estimate is referenced to the one before it, small per-observation errors compound and position drift grows without bound over a flight. A system that is both self-contained and free of drift must therefore fix its position against something external and fixed, without depending on a transmitted signal.

We introduce a drone that does not require radio signaling, long awkward fiber-optic cables, or heavy-duty hardware demands. Our drone uses an onboard companion computer with pre-downloaded georeferenced satellite imagery and a live camera feed to display an accurate VIO system with no error drift. The result is a system that is at once self-contained, like odometry, and absolute, like GNSS, without dependence on an external signal.

The objectives of this work were to establish whether a single downward-facing camera, combined with pre-flight downloaded satellite imagery, can produce absolute position fixes without any GNSS or radio link; and to design and manufacture a mount that carries a companion computer and camera on an unmodified airframe while isolating both from motor-induced vibration. The remainder of this report follows that structure. Section 3 develops the geometric basis of the approach, Section 4 documents the mechanical design and vibration isolation of the Prototype 1 payload mount, Section 5 describes the onboard software pipeline, and Section 6 presents the flight-test results and the limitations identified during testing.

3 System Concept

3.1 Navigation Systems

There are many popular navigation methods, such as GPS, SONAR, RADAR, LIDAR, and INS. However, each system individually comes with its own limitations and their cost rises steeply with accuracy. Small-format LIDAR units that are used for this size of aircraft are typically limited to roughly 100–300 m, GNSS and GPS receivers can be easily spoofed or jammed, and an INS is very susceptible to drift error.

3.2 The Problem

Conventional drone navigation depends on GNSS. When GNSS is jammed, spoofed, or obstructed, the aircraft loses absolute position. Visual Odometry can estimate an aircraft's relative motion but can drift without bounds since each estimate is based on a previously made estimate.

Our system instead calculates an absolute position fix from an individual camera frame. Every captured frame is registered against previously downloaded and georeferenced satellite imagery tiles. Since every position fix is attached to the respective map tile, errors do not accumulate over a flight. The estimated error for a 1-minute flight is the same for that of a 1-hour flight. Limitations for this are only based on how much data is uploaded and how accurate the previous satellite imagery tiles are.

3.3 Geometric Idea

The flight camera and satellite imagery face downward and create similar views of the same plane. Terrain relief and tall structures introduce a parallax displacement of approximately $(\Delta\ h/h)d$, which the model cannot represent. Every ground point in a frame is assigned a height value of zero which remains valid at an operating altitude of 50 meters and is what permits the homography. The homography itself is the transformation matrix that relates the two separate viewpoints of the same plane describing rotation, translation, scale, and perspective. The relation between points in homogeneous coordinates of both views is written as $\tilde{x}' \sim H\tilde{x}$ where $\tilde{x} = (u, v, 1)$ are the coordinates for a point in its corresponding frame and $\tilde{x}' = (\tilde{u'}, \tilde{v'}, w')$ are the coordinates for the same point in the mosaic. However, u' and v' are homogeneous and the coordinates in pixels are recovered by $u' = \frac{\tilde{u'}}{w'}$ and $v' = \frac{\tilde{v'}}{w'}$. The following is the 3x3 homography matrix:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

The top left 2x2 block describes values for rotation, scale, and shear. The first and second rows of the third column describe the planar translation. The bottom row of the first and second column describe the difference in perspective while the bottom right entry is overall scaling normalized to 1. Expanding $H\tilde{x}$ produces

$$\tilde{u'} = h_{11} \cdot u + h_{12} \cdot v + h_{13} \qquad \tilde{v'} = h_{21} \cdot u + h_{22} \cdot v + h_{23} \qquad w' = h_{31} \cdot u + h_{32} \cdot v + h_{33}$$

Dividing $\tilde{u'}$ and $\tilde{v'}$ by w'provides the two independent equations describing the location of the point in the mosaic in pixels.

$$u' = \frac{h_{11} \cdot u + h_{12} \cdot v + h_{13}}{h_{31} \cdot u + h_{32} \cdot v + h_{33}} \quad v' = \frac{h_{21} \cdot u + h_{22} \cdot v + h_{23}}{h_{31} \cdot u + h_{32} \cdot v + h_{33}}$$

Since it is only defined up to scale, the homography has 9 entries but only 8 of them are independent parameters, reducing degrees of freedom to 8. Each point correspondence supplies two equations, one for each coordinate. For an 8 DOF problem and 2 equations, the correspondence minimum is 4, provided that no three are collinear.

Our reference imagery comes from multiple public map providers where the world is divided into $2^z \times 2^z$ tiles at zoom z. Sourcing the correct tiles for our application is determined by matching the ratio of frame and tile ground sampling distance (GSD).

$$GSD_{frame} = \frac{h}{f_x} \quad GSD_{tile} = \frac{C \cdot cos\phi}{T \cdot 2^z}$$

The lens focal length, f_x, and tile length, T, are fixed.To have equal GSD values, we use zoom 20 tiles and fly at an altitude, h, of 50 meters. After matching map and frame GSD, at our operating altitude, the tile resolution is 0.1136 m/px, which presents another issue. With our resolution, the $256 \times 256px$ has a $29.09 \times 29.09m$ ground footprint while the frame has a $73.8 \times 41.29m$ ground footprint meaning a single tile covers 27.7 percent of the frame. To fix this we create a tile mosaic of nine tiles that is loaded on launch centered around our flight origin.

With the mosaic assembled and its geographic bounds known, any matched pixel within can be converted to a ground coordinate. The pixels position within the mosaic is interpolated across those bounds to give latitude and longitude:

$$\lambda = \lambda_L + \frac{P_x}{W_m}(\lambda_R - \lambda_L) \qquad \phi = \phi_T - \frac{P_y}{H_m}(\phi_T - \phi_b)$$

The latitude and longitude of the pixel are then projected to a local East-North frame in meters, referenced from the flight origin:

$$E = (\lambda - \lambda_0)\frac{\pi}{180}R_E \cos\phi_0 \qquad N = (\phi - \phi_0)\frac{\pi}{180}R_E$$

Because these world coordinates carry metric units, the homogpraphy fitted from them to the camera image keeps those units, and the resulting translation is in meters. To recover position and orientation of the camera, a projection model maps a 3D world point to a pixel using the cameras orientation, its displacement, and its fixed calibration.

$$\tilde{x} \sim K \begin{bmatrix} R \mid t \end{bmatrix} X$$

Where X is a 4×1 matrix of a ground point in homogeneous coordinates and $\begin{bmatrix} R \mid t \end{bmatrix}$ is a 3x4 matrix with the first three columns r_1, r_2, and r_3 describing the worlds East, North, and vertical axes as seen from the camera and the fourth column t describes the world origin relative to the camera. Using the previously stated planar assumption, the product $r_3 Z$ vanishes for every point. What remains is a 3×3 matrix acting on $(X, Y, 1)^\top$, the homogeneous coordinates of a point on the ground plane, which is by definition a homography:

$$H = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

Under the planar assumption, the camera projection is a homography, and its columns are the first two columns of the camera's rotation together with its translation.

4 DJI Phantom 3 Mounted System

4.1 Purpose

The purpose of this prototype was to create a cheap platform to test and develop our approach and software. Modifying the existing DJI Phantom allowed us to test the system in a very time and cost-effective manor. The main components of this design were a pre-owned DJI Phantom 3 as the mode of flight, an NVIDIA Jetson Orin Nano as the main computer, and an Arducam IMX477 camera for capturing images.

4.2 Process

4.2.1 Material Selection

After strength-to-weight optimization, our final 3D print was made of carbon fiber reinforced nylon 12. We used SOLIDWORKS simulations to acquire the stress analysis of our designs and test different materials such as PLA, ABS, PC, and Nylon 12 Carbon Fiber (PAHT-CF). PAHT-CF is relatively cheap, lightweight, and created a very rigid frame, which helped maintain the target natural frequency for reducing vibration to the computer and camera.

4.2.2 Mechanical Design

The DJI Phantom 3 is not meant to be modified, and every part of its design is specific to the orientation and weight of the factory parts. This proved to be a challenge, as adding 300 grams of extra weight would significantly throw off the in-flight performance since the flight controller is pre-programmed and could not be altered. We decided to remove the camera and gimbal to reduce the starting weight by 170 grams, which provided the mounting screw holes and space to add our navigation system. When designing the mount for our components, the three main constraints were weight, strength, and natural frequency. To determine the maximum payload that the DJI could handle in flight, we designed a dead load mount.

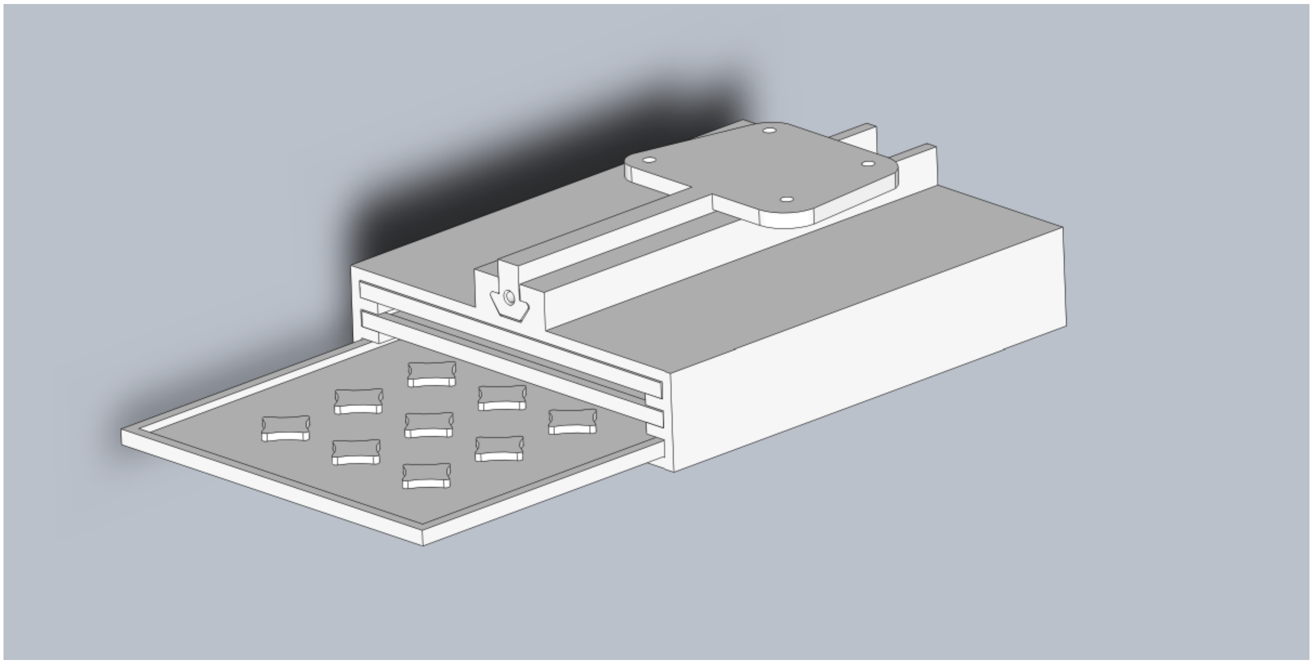

Figure 1: Coin-slot SolidWorks assembly

The geometry and location on the drone were made to replicate where our real system would sit. Using quarters on separate trays allowed us to incrementally increase the load between flights. Our discovered maximum additional weight where we still had decent flight stabilization was 560 grams. We optimized the parts to be as light as possible within our bounds to maintain agility. To solve for vibration, we calculated isolated motor and blade pass frequencies to allow for clear image capturing.

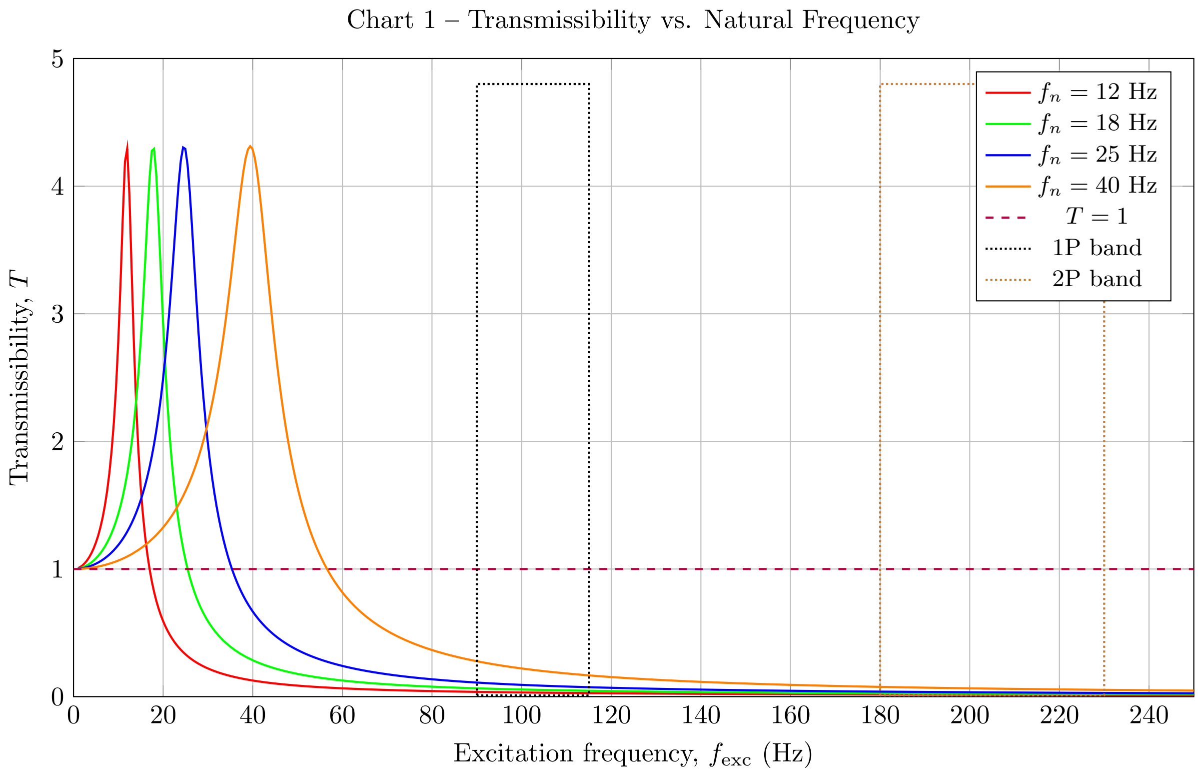

Figure 2: Effect of isolator natural frequency at a damping ratio of $\zeta = 0.12$.

We used chart 1 to determine the natural frequency of our mount with a constant damping ratio. The 1P and 2p rectangles are the regions where the motors idle and maintain a hover. A higher natural frequency pushes the resonance hump closer to the idle RPM, while also fighting natural movements. A lower natural frequency would isolate more at lower excitation frequencies, but would have little resistance to sway during in-flight maneuvers. We decided to target a natural frequency of 18 Hz.

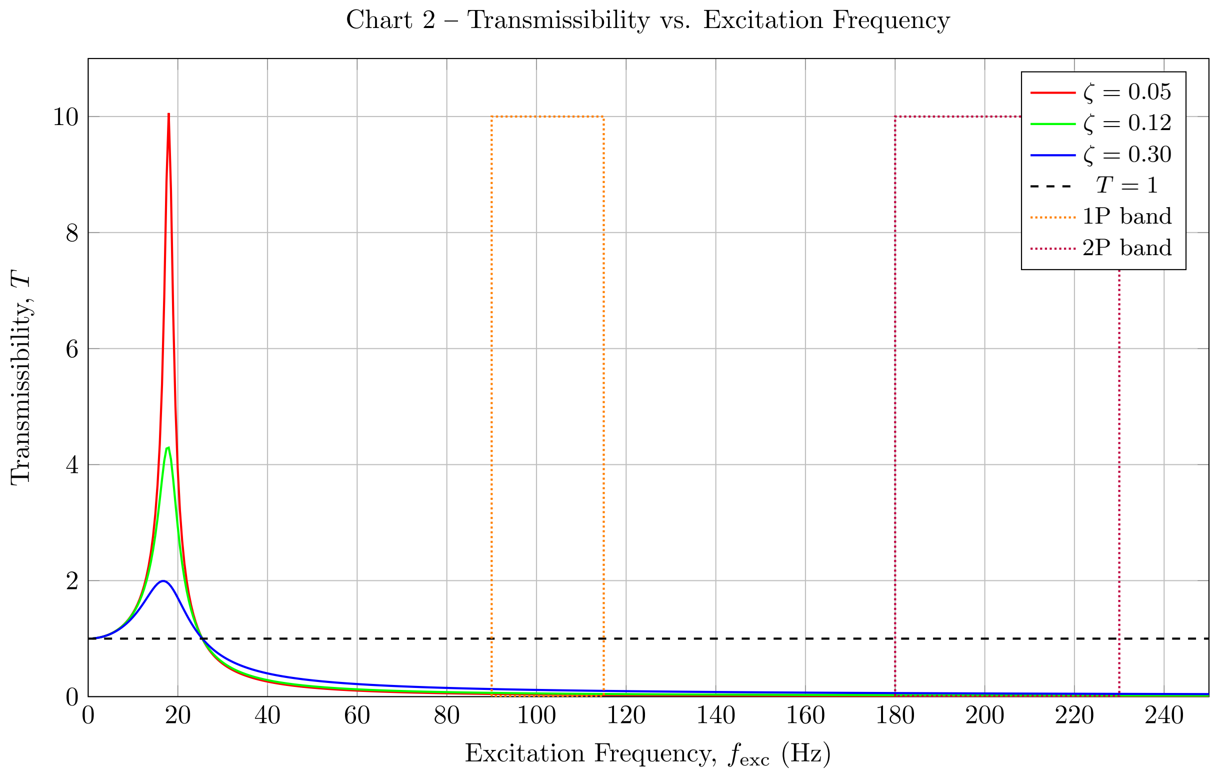

Figure 3: Effect of damping ratio on vibration transmissibility for $f_n = 18$ Hz.

Chart 2 plots how much of the airframe's vibrations reach our mount across the excitation frequency spectrum for three different damping ratios. This was used to determine the damping material. A material with a lower damping ratio has a much higher resonance hump but isolates more in the 1P and 2P bands. Since we are using the drone most frequently in the 150-180 Hz range for consistent hovers, we chose a damping ratio around 0.12.

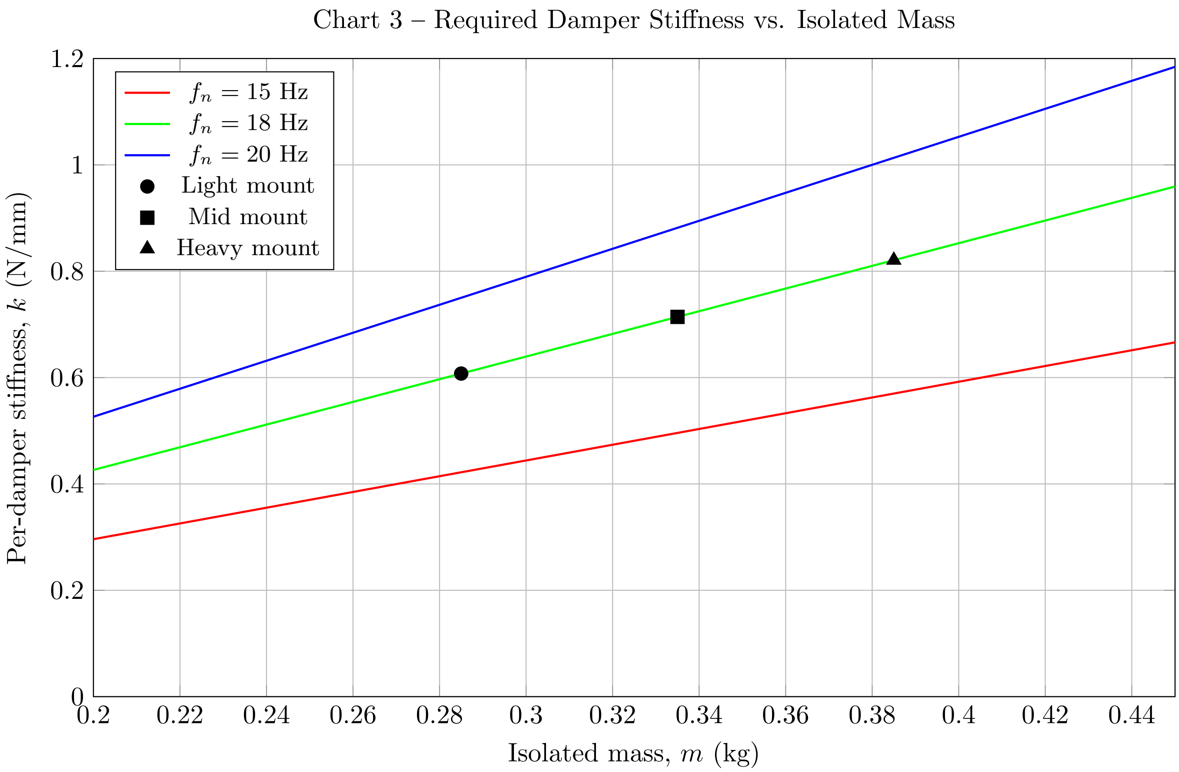

Figure 4: Required stiffness for each of the six dampers as payload mass changes.

Chart 3 was used to determine both the damper stiffness and our target weight. Optimizing for lighter weight while using a hexagonal 6-damper pattern, our mass target was 285 grams with a damper stiffness of 0.6 N/mm.

Using all of the chart data collectively to obtain our specific design constraint values provided a straightforward design process. After dimensioning the pre-existing screw holes on the DJI, we were able to design the main mounting plate that would support the weight of the Jetson and camera. After the dimensions and first iteration were made, we started a series of static force simulations to determine required changes to minimize the maximum displacement under a 4x gravity loading condition. This was our worst-case load used to simulate a rough landing.

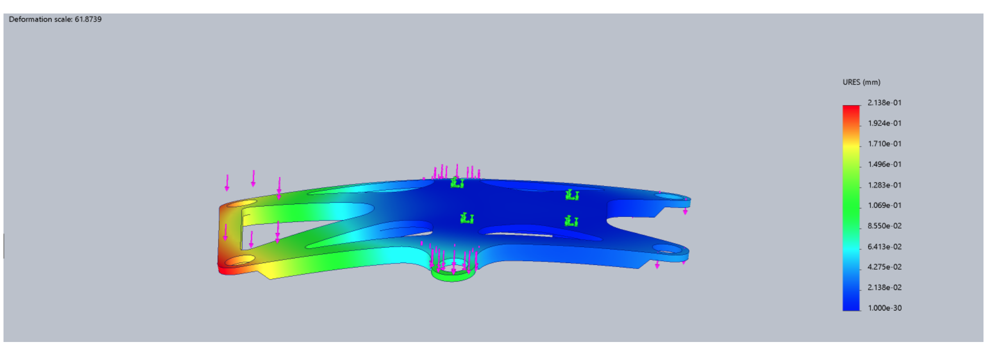

Figure 5: SolidWorks static force analysis with 15 newtons total applied load

We then created a housing for the Jetson, which provided separate mounting points for the camera.

With the complete and final assembly satisfying all of our previous criteria, we began printing the parts. All

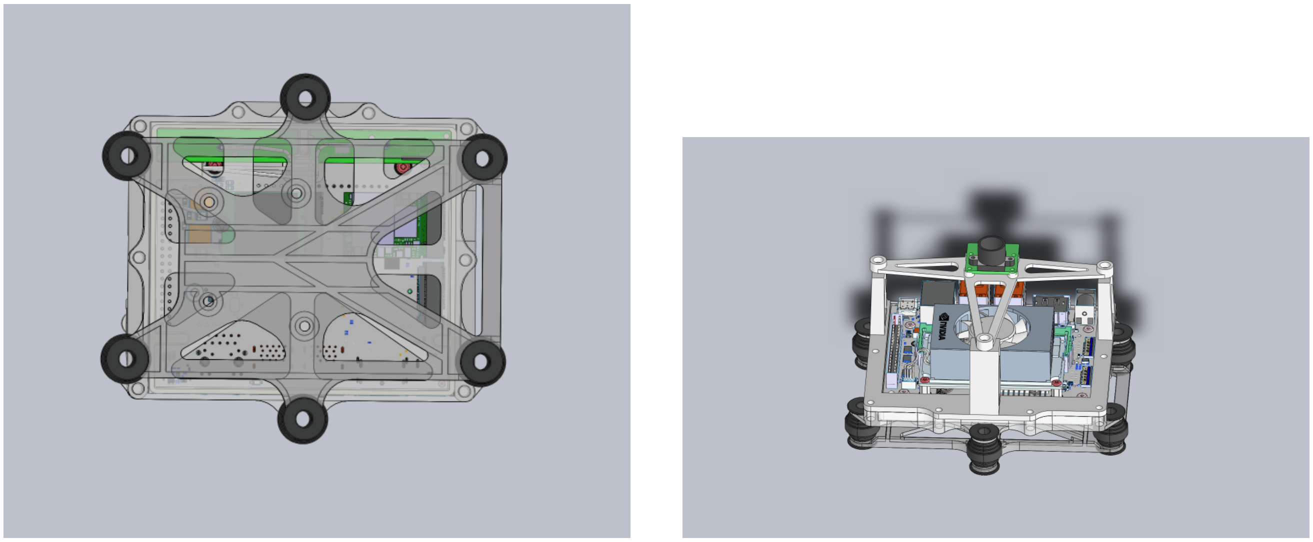

Figure 6: Final Jetson and IMX477 mount with vibration dampers

of the connections on the Jetson plate used M3 screws and brass heat-set threaded inserts. The camera strut was mounted with M2 screws and inserts, with all threads secured with Locktite to survive frequent vibration during flight.

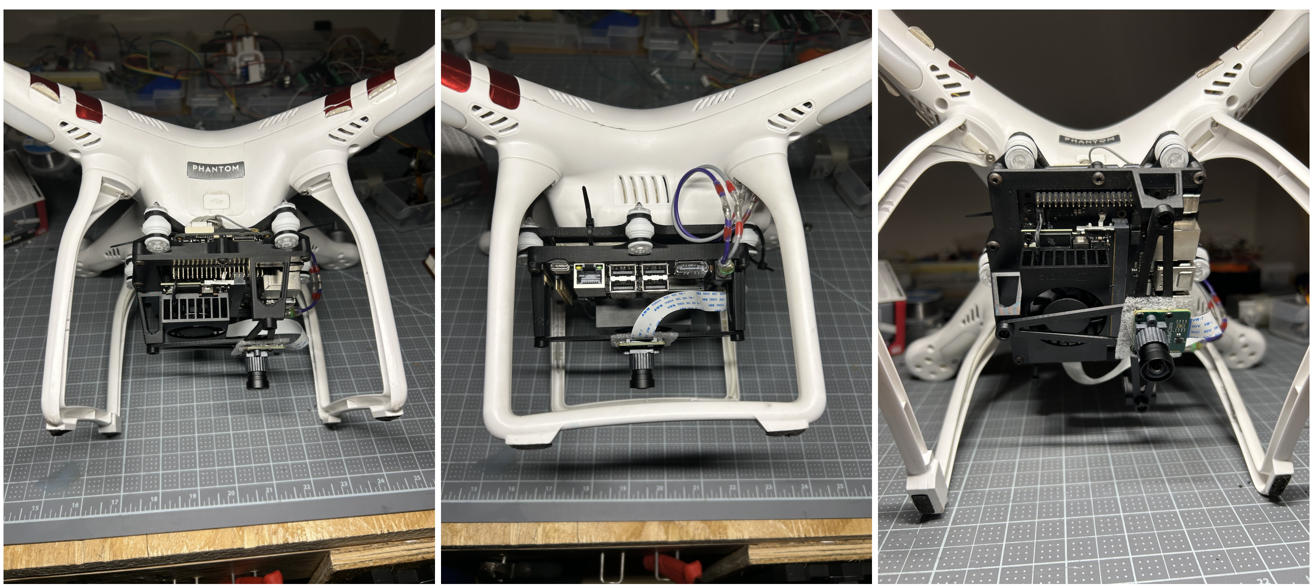

Figure 7: Final DJI modifications implemented

5 General Software Pipeline

Section 3.3 established the geometry that makes a position fix possible, this section describes the software carried out on the Jetson during flight, and the settings we used. Every candidate frame experiences the same stages in order, image aquisition, correction, reference preparation, detection, matching, filtering, pose recovery, and validation.

The ground-facing CSI camera captures raw Bayer sensor data that must be converted into a usable image. This is performed by the Jetson hardware through NVIDIA's Argus library. After a frame is passed, it goes through three operations. First, the frame is geometrically corrected by cropping the lens distortion by pulling the pixels inward from the frame edges. The extra gray area in the frame is cropped away while narrowing the field of view. This operation is important since the homography model assumes a pinhole camera. The second operation is reducing the resolution. The sensor resolution 3840×2160 correlates to 1280×720 camera output which is scaled down to 640×360 for each processed frame. The processed frame cannot be chosen for speed alone. It is bounded by the requirement that the frame's GSD stay near the tile's in order to resolve the features contained in the mosaic. Scaling down the image lets us optimize the first operation by combining the distortion correction and resolution reduction into a single table that is built once at startup. This table is referenced for each frame after and lightens the load of preprocessing. The third operation is normalizing the contrast. Since our aerial imagery can contain bright sunlight spots and dark tree lines, we cannot use a global adjustment. Using Contrast Limited Adaptive Histogram Equalization (CLAHE), the image is divided into an 8x8 grid, and a histogram is computed for each segment individually.

At the same time as the image processing, the georeferenced satellite imagery tiles covering the origin at takeoff and the eight surrounding tiles are loaded from storage and stitched together. The mosaic stitching matters for the same reason stated in section 3.3. Both the tiles and camera frame receive identical CLAHE to help account for different times of day and seasonal differences. The descriptor tolerates a uniform brightness change, but CLAHE is neither uniform nor global, requiring both images to undergo the same normalization.

After a frame is processed and a mosaic is loaded, the feature detection node runs. Using Oriented FAST and Rotated BRIEF (ORB) feature detection, both images are reduced to sets of keypoints, each carrying a descriptor that encodes the appearance of the image patch around it. Detection is by the FAST test: a candidate pixel is compared with a ring of sixteen pixels surrounding it, and a corner is declared when at least nine consecutive ring pixels are either all brighter or all darker than the center using the OpenCV default threshold of 20. Using a scale pyramid for scale invariance with a scale factor of 1.25 over ten levels, the FAST corner detector runs over each level of the pyramid independently. Each keypoint is then assigned an orientation taken from the intensity centroid of its patch, that is, the direction from the center of the patch to its center of mass. The descriptor is a rotated BRIEF string using the default 256 bit length: pairs of pixels within the patch are compared in brightness, one bit recorded per comparison, with the sampling pattern first rotated by the keypoint's orientation. Rotating the pattern is what makes the descriptor independent of the aircraft's heading, and means that the matching compares patches of appearance rather than the corner pixels themselves. For every feature in the live frame, the two most similar features in the reference are found, similarity being the Hamming distance between the two binary descriptors. Each camera frame retains 2000 features while the mosaic retains 5000. The mosaic is allocated more features than the frame since it covers nine times the ground area. The raw matches are then passed to the filtering node, which first applies Lowe's ratio test. A match is only accepted if the best candidate is significantly better than the runner-up, that is, if the ratio of the best distance to the second-best falls below a fixed threshold of 0.75. Two equally good candidates indicate a repeated texture, such as tree canopy or a row of similar rooftops, and the match is discarded as ambiguous. If it passes the ratio test, then it moves on to RANSAC geometric verification. RANSAC tests whether the matches agree on a single consistent transform. Four correspondences are drawn at random, the homography they imply is computed, and every remaining match is re-projected through it; if matches agree and fit within the pixel tolerance, they are counted as supporting the transform hypothesis. The draw is repeated many times and the hypothesis with the largest support is kept. The homographty estimated during this filtering step serves only to identify which matches to keep, and is then discarded. The pose is recovered from a separate homography fitted from ground coordinates as described in Section 3.3.

The surviving correspondences are passed to the pose recovery described in Section 3.3. Each recovered position then must pass validation gates. First a reprojection check, the recovered pose is used to predict where its own supporting points should have appeared and compared against where they were measured using a reprojection tolerance of 5 px. Then physical plausibility gates confirm whether the calculated fix is possible. From the fix the altitude and tilt from nadir must fit within a valid range determined by physical constraints. The position is then published and sent back to update the estimated altitude used for scale matching and to select the next reference tile region to load.

6 Discussion

6.1 Related Work

6.2 Results

The final modifications, shown in Figure 7, have performed seven flights and withstood several hours of flight time without damage or loosening. Our vibration damping system allowed us to capture great frames and supply our ROS environment with clear images. The total weight of the mount did not affect the agility of the Phantom in any noticeable way and has outperformed expectations in terms of rigidity and strength. After two months of bench and field tests, we obtained our first accurate position. We were able to accomplish this with multiple script changes and many launch parameter edits. Our best results came when we limited the overall gap between satellite images and camera frames by sourcing the most up-to-date tiles and flying in locations with clear measurable features. The last two of our seven flights produced the most accurate positions within 10 meters of our flight origin.

Across the seven flights the system produced 13 successful camera-to-satellite match events. At the 50 m operating altitude the camera ground sample distance was 0.115 m/px, closely matching the 0.1136 m/px of the zoom-20 satellite tiles used as the reference, and the ROS 2 pipeline processed live frames against 3×3 tile mosaics in real time. For those final two flights the mean horizontal error was 8 m, measured against the aircraft's onboard GNSS receiver, whose own accuracy is specified at ± 3 m.

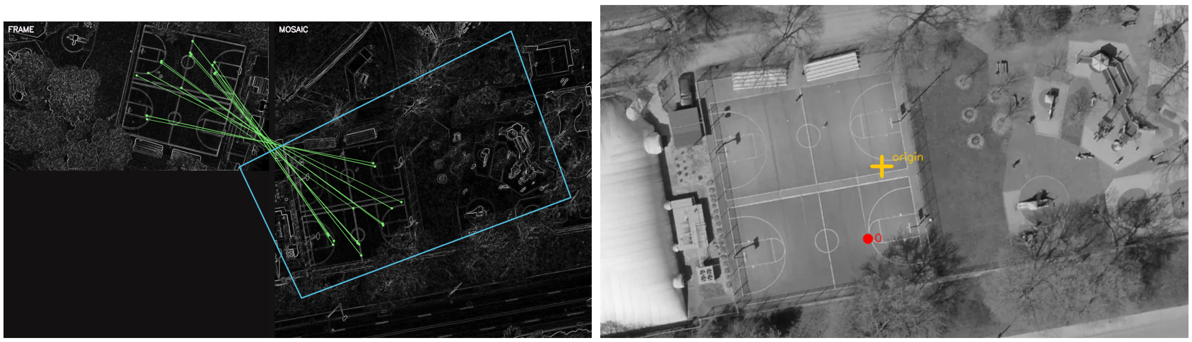

Figure 8: Mellon Park flight results. System view on the left, with inliers in green and frame footprint in blue. In-flight calculated position with map tile background on the right, origin in orange and position marked in red.

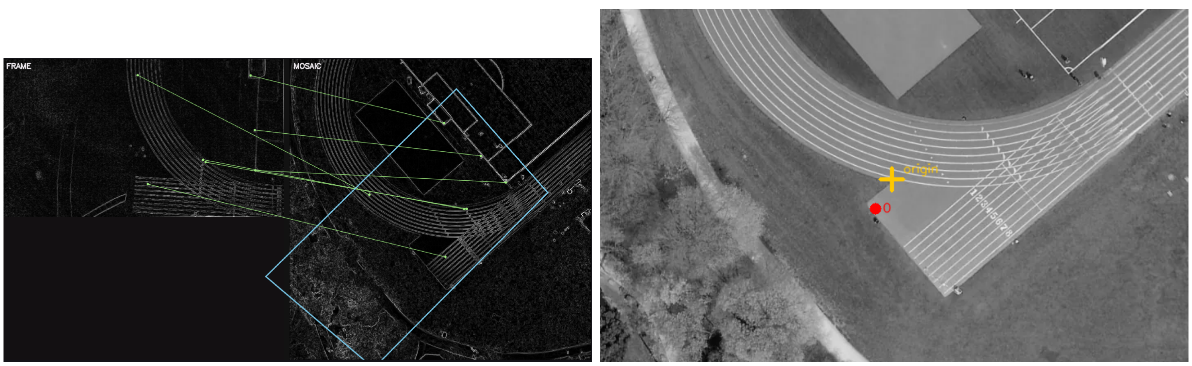

Figure 9: Schenley Park flight results. System view on the left, with inliers in green and frame footprint in blue. In-flight calculated position with map tile background on the right, origin in orange and position marked in red.

6.3 Limitations

The most impactful limitations were the lack of sensors and other hardware, as well as the hard altitude limit of 50 meters that the DJI was capable of. With only a camera and no usable IMU, all the 8-DOF homography had to be estimated from image correspondences alone, and the camera's 6-DOF pose was recovered from it without any independent estimate of attitude or scale. Since the IMU can provide orientation, it would help determine the drone's pitch and yaw, which would benefit ORB feature detection and homography. The hard altitude limit of 50 meters made it difficult to obtain the correct zoom tiles. At 50 meters, the ground sampling distance is 0.1136 meters per pixel, which requires a true zoom of 20 tiles. Commercial z20 tiles are frequently unsampled from coarser source imagery, so the nominal 0.1136 m/px may not represent true captured detail. This also prevented us from testing the system at different altitudes, which would have been helpful for determining performance and accuracy with a larger frame footprint.

6.4 Future Work

After collecting all of this data and learning a lot about ROS architecture, computer vision methods, feature detection and description, and quadcopter flight dynamics, we have many proposed upgrades for a full VIO map-to-frame matching navigation system. We have already begun planning the second prototype, which will include many new sensors to improve it. Including a range finder for automatic tile scaling, multiple IMU for aircraft orientation, and a two-axis camera gimbal. These three additions should greatly improve the efficiency and robustness by limiting the homography to a 2D transformation.

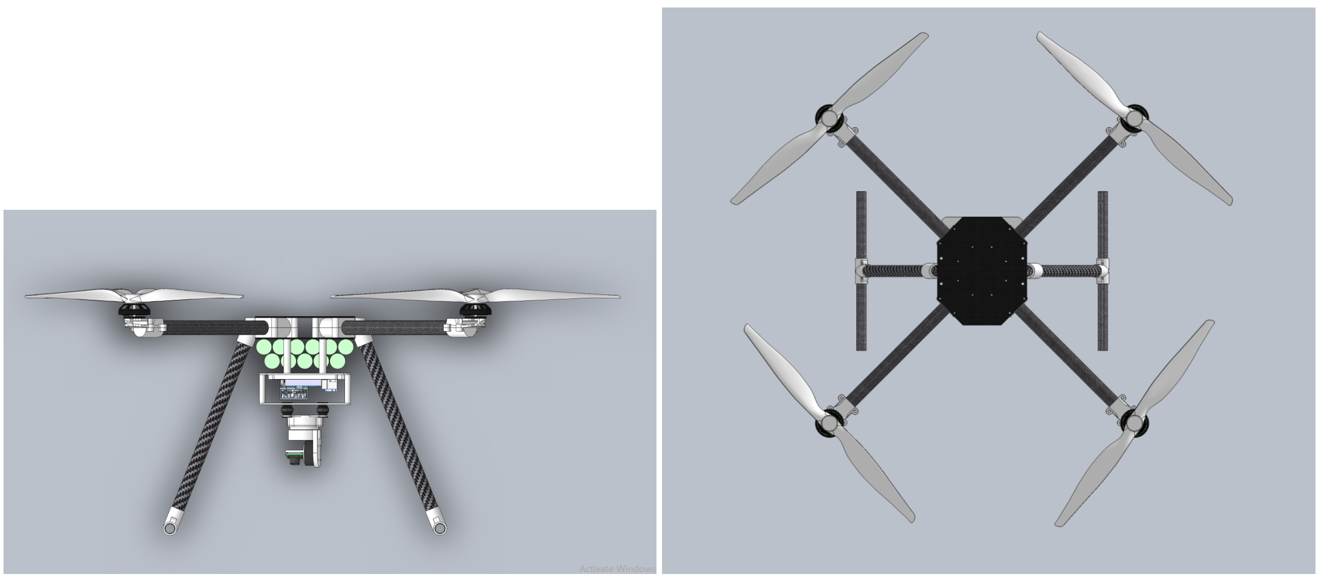

Figure 10: Prototype 2 SolidWorks assembly

7 Conclusion

We implemented a GNSS-independent localization system and flight-tested it on a modified DJI Phantom 3. Removing the factory camera and gimbal reduced airframe mass by 170 g, and the incremental dead-load testing established a maximum additional payload of 560 g for stable flight. The final mount was printed in carbon-fiber-reinforced nylon 12 (PAHT-CF) and carried an isolated mass of 285 g on six dampers of 0.6 N/mm each, giving a design natural frequency of 18 Hz at a damping ratio of 0.12. Static analysis at an 11.2 N loading condition produced a maximum displacement of 0.2138 mm.

The ROS 2 pipeline onboard processed live camera frames against 3×3 georeferenced tile mosaics in real time. At the 50 m operating altitude, the camera ground sample distance was 0.115 m/px, matching the 0.1136 m/px of the zoom-20 satellite tiles used as the reference. Across seven flights, the system produced 13 successful camera-to-satellite match events. The final two flights returned absolute position fixes within 10 m of the flight origin, with a mean horizontal error of 8 m measured against the aircraft's onboard GNSS receiver, whose own accuracy is specified at ± 3 m.

8 References

[1] YTTEK Technology Corp., "What is Drone Communication," *YTTEK Insights*, Nov. 4, 2025.

[2] Leher AG, "AI in Drone Technology: Benefits and Challenges," *Leher Blog*, Aug. 11, 2025.

[3] B. Bollard, A. Doshi, N. Gilbert, C. Poirot, and L. Gillman, "Drone Technology for Monitoring Protected Areas in Remote and Fragile Environments," *Drones*, vol. 6, no. 2, art. 42, 2022.

[4] B. Hahn, "Drone Mapping 101: A Beginner's Guide to Capturing Accurate Geospatial Data," *Birdi Blog*, May 14, 2026.

[5] Doodle Labs LLC, "What is Drone Jamming and How Can It Be Countered?" *Doodle Labs Blog*, Dec. 2024.

[6] Federal Aviation Administration, "Recreational Flyers & Community-Based Organizations," U.S. Department of Transportation, Mar. 18, 2026.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2004.

[8] H. Goforth and S. Lucey, "GPS-denied UAV localization using pre-existing satellite imagery," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 2974–2980.

[9] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems IV*, P. S. Heckbert, Ed. San Diego, CA, USA: Academic Press, 1994, pp. 474–485.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Barcelona, Spain, 2011, pp. 2564–2571.

[11] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conf. Computer Vision (ECCV)*, Graz, Austria, 2006, pp. 430–443.

[12] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. European Conf. Computer Vision (ECCV)*, Heraklion, Greece, 2010, pp. 778–792.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[15] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, art. eabm6074, 2022.