

Low-Cost Strapdown Stellar-Inertial Positioning for High-Altitude Platforms

Julius Arolovich, Seungyeon (Rachel) Lee, Eddie Zhang
Electrical and Computer Engineering, Carnegie Mellon University

Abstract—Small UAVs operating in GPS-denied or GPS-degraded environments lack a reliable source of absolute position, causing inertial navigation errors to grow unbounded over time. Prior approaches utilize visual positioning during daytime by comparing live perception against a known imagery prior, though this does not readily extend to the nighttime case. We present a lightweight celestial positioning module that restores absolute localization at night by observing stars with an upward-facing camera. The system identifies stars in captured images, matches them against a known star catalog to determine where the camera is pointing, and converts this to a geodetic latitude and longitude using precise time and the Earth rotation model. Each observation simultaneously refines the platform’s attitude estimate and corrects accumulated position drift from inertial sensors. The module targets ≤ 10 km global position accuracy, sub-second pipeline latency, and a total mass under 500 g in order to be tractable for use with small UAVs.

Index Terms—Celestial navigation, GPS-denied localization, plate solving, sensor fusion, unmanned aerial vehicle

1 INTRODUCTION

Global Navigation Satellite Systems (GNSS) provide the primary source of absolute position for most unmanned aerial vehicles. However, GNSS signals are vulnerable to jamming, spoofing, and environmental occlusion, and are unavailable in contested or remote operating areas. When satellite positioning is lost, a UAV must rely on inertial dead reckoning, which accumulates drift without bound due to accelerometer and gyroscope bias. Over minutes of flight, this drift can grow to kilometers, making positioning uninformative to autonomous or manned operation.

Daytime visual positioning methods partially address this gap by matching live camera imagery against georeferenced satellite or terrain databases. These approaches depend on illuminated, feature-rich scenes and a prior imagery dataset, conditions that break down at night when surface features are largely invisible. Nighttime operation is operationally important; many surveillance, search-and-rescue, and logistics missions are conducted after dark, yet few lightweight positioning solutions exist for this regime.

Celestial navigation offers a different observation source. The positions of stars on the celestial sphere are cataloged to sub-arcsecond precision and are visible from any location on Earth with a clear sky, independent of

ground infrastructure or RF signals. By identifying stars in an image and matching them against a known catalog, a system can recover its pointing direction on the celestial sphere and, given accurate time, convert that fix to geodetic latitude and longitude via known relationships.

This project develops a lightweight celestial positioning module designed to mount on a small UAV. The module uses an upward-facing camera to capture nighttime sky images, a star detection and plate-solving pipeline to determine absolute pointing, and a sensor fusion architecture that combines these visual fixes with inertial measurements from an onboard IMU. The goal of the project is a self-contained system that provides absolute position estimates in simulated GPS-denied nighttime conditions while targeting sub-10 km accuracy and a total mass under 500 g.

2 USE-CASE REQUIREMENTS

To support our product use case, the system must meet several quantitative performance requirements.

First, the system shall estimate global position (latitude and longitude) with an error no greater than 10 km under clear nighttime sky conditions. This level of accuracy is sufficient for navigation recovery and correction to be later used by a Kalman filter.

Second, the system shall produce a position estimate within 1 second of capturing an image. This ensures timely localization updates while remaining computationally feasible for our hardware.

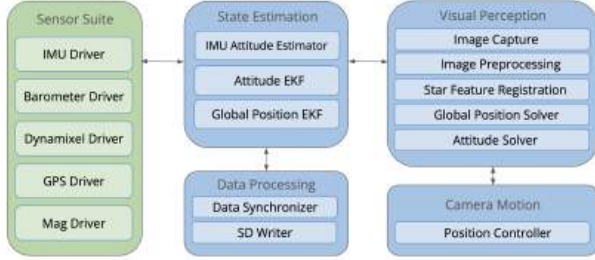
Third, the system shall operate under nighttime outdoor conditions with 90% or greater visibility of celestial features. Since celestial navigation relies on star observations, nighttime operation is a fundamental requirement of the system.

Fourth, the system shall provide improved localization accuracy compared to inertial dead reckoning alone over equivalent time periods when compared to gps at over 95% of the visual update. This ensures that the celestial positioning subsystem provides measurable benefit by correcting accumulated inertial drift.

Finally, the system shall be lightweight (≤ 500 g) and mechanically compatible with small UAV platforms, ensuring deployability without requiring specialized aircraft.

3 ARCHITECTURE AND PRINCIPLE OF OPERATION

The system consists of three principal subsystems: a *visual solving pipeline* that converts raw sky images into absolute celestial measurements, a *sensor interface layer* that collects inertial and auxiliary data, and a *state estimation engine* that fuses both streams into a continuous position and attitude estimate. Fig. 3 shows the top-level block diagram; the remainder of this section introduces each subsystem.



3.1 Visual Solving Pipeline

The visual pipeline accepts grayscale images from the upward-facing IMX477 camera and produces two outputs: a boresight direction expressed as right ascension and declination (RA/Dec) on the celestial sphere, and an attitude estimate in the form of a body-to-world rotation matrix.

The underlying principle is *plate solving*, a technique originating in astrometry. Stars are point sources whose positions on the International Celestial Reference System (ICRS) are cataloged to sub-arcsecond precision. When the camera captures an image of the sky, detected star centroids form a geometric pattern that is unique to that particular patch of the celestial sphere. The plate solver identifies this pattern by matching triangles or quads of detected sources against a precomputed index of catalog star groupings. A successful match yields a World Coordinate System (WCS) transformation that maps every pixel in the image to an RA/Dec coordinate. Evaluating this transformation at the image center gives the camera boresight direction.

With the aid of camera intrinsics made available through conducting camera calibration with the sensor, the pipeline additionally recovers the full three-axis attitude of the camera. Each detected star's pixel position is projected into a unit vector in the camera frame using the pinhole model and the calibrated intrinsic matrix. The corresponding sky-frame unit vector is obtained from the WCS. The Kabsch algorithm then finds the rotation matrix $\mathbf{R}$ that best aligns the two sets of vectors via singular value decomposition, yielding the camera's orientation relative to the celestial frame.

The point directly overhead (the zenith) has a declination equal to the observer's latitude and a right ascension equal to the local sidereal time. Since the local sidereal time differs from the known Greenwich Mean Sidereal

Time (GMST) by exactly the observer's longitude, a zenith RA/Dec measurement and an accurate clock are sufficient to recover both coordinates:

$$\varphi = \text{Dec}_{\text{zenith}} \quad (1)$$

$$\lambda = \text{RA}_{\text{zenith}} - \text{GMST} \quad (2)$$

When the camera is tilted away from the true zenith, the plate-solved boresight RA/Dec must first be corrected using the attitude estimate from the Kabsch alignment before applying Eqs. (1)–(2). This is why accurate attitude estimation is directly required for position recovery. Timing accuracy is of equal importance: the Earth rotates at $\sim 15^\circ/\text{hr}$, so each second of clock error introduces roughly 28 km of longitude error at the equator. We satisfy this by syncing the RPi clock to a remote time server, or GPS if available, before falling back to the board's internal oscillator.

3.2 Sensor Interface Layer

The sensor interface layer manages all hardware peripherals and provides timestamped data streams to the state estimator over ROS2 topics.

The IMU provides three-axis accelerometer and gyroscope measurements at 100 Hz over I2C. These measurements are expressed in the body frame and include both the translational acceleration of interest and the gravity vector, which must be separated using an accurate attitude estimate.

The GPS module provides NMEA position fixes and a PPS timing signal over UART. During development and validation, the GPS stream serves as ground truth. In operational, the GPS is unavailable and the system clock free-runs on the RPi oscillator.

The barometer provides pressure-derived altitude estimates over I2C - functionally an independent vertical constraint to the state estimator.

3.3 State Estimation Engine

The state estimator fuses the high-rate inertial stream with the lower-rate visual observations using a dual-filter architecture, illustrated in Fig. 3.

The first filter is an *attitude estimator* that maintains the body-to-world rotation matrix $\mathbf{R}_{wb}$. At high rate, the gyroscope angular velocity measurements propagate the orientation forward in time through integration. This propagation accumulates drift due to gyroscope bias, so whenever a visual observation arrives, the attitude derived from the Kabsch alignment provides an absolute correction that resets the drift.

The corrected rotation matrix serves a large role in the second filter by transforming accelerometer measurements from the body frame into the world frame and removing gravity. Without accurate attitude, the gravity vector is incorrectly compensated, introducing acceleration errors that grow quadratically into position drift.

The second filter is a Extended Kalman Filter (EKF) that estimates three-dimensional position and velocity. Between visual updates, the EKF propagates the state forward using the gravity-compensated acceleration from the attitude estimator. When a visual position fix becomes available, the EKF incorporates it as a measurement update, correcting accumulated drift in both position and velocity.

This separation of attitude and position estimation into two coupled filters provides several architectural advantages. The attitude filter operates at the IMU rate (100 Hz) while the position filter updates at the visual cadence (approximately 1 Hz), avoiding the performance difficulties of achieving a single filter spanning two entirely different time scales. Each filter also has a smaller state dimension, reducing computational cost on the RPi 5 and simplifying tuning and introspection into each independent subsystem of the state engine.

3.4 System Data Flow

The overall data flow proceeds as follows. The camera captures an image of the sky. The visual pipeline detects stars, solves for the plate, and extracts boresight RA/Dec and attitude. The boresight is converted to geodetic coordinates using the current timestamp and the Earth rotation model. The attitude observation corrects the gyroscope-propagated orientation in the attitude filter. The corrected orientation is used to transform accelerometer data for inertial propagation in the position EKF. The geodetic position observation corrects accumulated inertial drift. The cycle repeats at each servo step, producing a continuous position estimate from a combination of star features and inertial sensing.

4 DESIGN REQUIREMENTS

The use-case requirements in Section 2 (≤ 10 km position error, < 1 s latency, < 500 g mass, nighttime GPS-denied operation) flow down into a set of quantitative design specifications for each subsystem. We focus here on the requirements that drive the visual solving pipeline, since this is where the dominant error sources in the position estimate enter the system.

4.1 Star Detection Sensitivity

The IMX296 is a global-shutter monochrome sensor, which directly affects this requirement in two ways. First, the global shutter eliminates the row-wise integration skew of a rolling shutter, which IMX477 uses, so star centroids remain geometrically consistent during platform motion. Second, the sensor captures infrared light, which improves sensitivity for detecting dim stars at short exposures.

4.2 Plate Solving Angular Accuracy

Since 1° on the celestial sphere maps to about 111 km on the ground (projected through the zenith), the 10 km use-case requirement gives us:

$$\varepsilon_{\text{boresight}} \leq \frac{10 \text{ km}}{111 \text{ km}/^\circ} \approx 0.09^\circ \quad (3)$$

The servo sweeps the camera through 360° , collecting up to 8 independent boresight measurements per revolution. Averaging these reduces error by roughly $\sqrt{8} \approx 2.8\times$, so the per-frame spec of 0.1° gives us enough margin to hit 10 km after fusion.

4.3 Field of View Specification

FOV sets the plate scale, which is just:

$$s = \frac{\text{FOV}}{N_{\text{pixels}}} \quad (4)$$

With the IMX296's 1456 horizontal pixels and a 56° FOV, we obtain approximately $0.038^\circ/\text{pixel}$ (~ 140 arcsec/pixel). This is well under the 0.1° boresight target while remaining wide enough to capture sufficient stars at mid-latitudes. Calibration must pin down the true focal length to $\leq 1\%$ so that the plate solver's FOV hint remains tight enough for fast convergence.

4.4 Compute Latency

The 1-second budget breaks down as: image capture (≤ 50 ms), star detection (≤ 100 ms), plate solving (≤ 700 ms), and boresight extraction and geodetic conversion (≤ 10 ms). Plate solving dominates this budget. On the RPi 4, `solve-field` from `astrometry.net` completes in 300–800 ms when given a good FOV constraint (searching $0.5\times$ to $2\times$ the nominal). Without that constraint, blind solves can take 10+ seconds, well outside the budget.

4.5 Timestamp Synchronization

The Earth rotates at $\sim 15^\circ/\text{hr}$, so a timing error Δt causes longitude error:

$$\Delta\lambda \approx 0.25^\circ \times \Delta t [\text{seconds}] \quad (5)$$

That's about 28 km per second of error at the equator. To stay under 1 km of timestamp-induced error we need $\Delta t \leq 36$ ms; we spec ≤ 50 ms for margin. We achieve this by syncing the RPi clock to GPS PPS before going GPS-denied, then free-running on the onboard oscillator.

4.6 Summary of Design Specifications

Table 1 collects the specifications derived above.

Table 1: Design Specifications Summary

Parameter	Spec	Unit
Min. stars detected	≥ 8	per frame
Centroid accuracy	≤ 0.5	pixels
Boresight accuracy	≤ 0.1	degrees
Plate solve latency	≤ 700	ms
Total pipeline latency	≤ 1000	ms
Timestamp sync error	≤ 50	ms
FOV calibration accuracy	≤ 1	% of true
Module mass	≤ 500	grams

5 DESIGN TRADE STUDIES

This section documents the three trade studies that most strongly shaped the system as built: the camera sensor selection, the choice of plate-solving engine, and the state-estimator architecture. Each study considers candidate options against the design specifications in Section 4, and the rationale documented here reflects the configuration we ultimately deployed.

5.1 Camera Sensor Selection

The dominant constraints on the imaging sensor are low-light sensitivity (so that short exposures still yield enough detected stars) and geometric stability of star centroids during platform motion. We evaluated two candidate Raspberry Pi-compatible camera sensors with official `libcamera` support: the IMX477 and the IMX296.

The IMX477's C/CS-mount compatibility made it our initial baseline; however, two limitations drove us away from it during integration. First, the IMX477 uses a rolling shutter, which integrates each row at a slightly different time. For an upward-facing camera that may swing through several degrees per second during normal flight, this row-wise time skew distorts star centroids along the scan direction and corrupts the geometric assumptions of the plate solver. Second, the IMX477 has a color sensor with a Bayer color filter array, which limits sensitivity when imaging dim stars at short exposures.

The IMX296 is a global-shutter monochrome sensor. The global shutter exposes all pixels simultaneously, eliminating the rolling-shutter distortion that affects centroid geometry under motion. The monochrome pixel array, combined with the IMX296's infrared sensitivity, improves star detection at short exposures. The trade is resolution.

We selected the IMX296 as the imaging sensor. The shutter type and monochrome/infrared sensitivity drove the decision. The resolution trade was bounded by the design requirement and was therefore not binding.

5.2 Plate Solving Engine: ASTAP vs. astrometry.net

Two open-source plate-solving engines are practical candidates for an embedded Linux platform: ASTAP and astrometry.net's `solve-field`. We initially selected ASTAP because it is faster than astrometry.net on raw solve time. However, ASTAP requires a full all-sky star

index loaded as a single package, which occupies several gigabytes on disk. This is impractical for an embedded deployment on the RPi 4.

Astrometry.net distributes its indices in sky-region fragments and accepts a pointing-region hint that restricts which indices it loads. For our operating latitude we can ship only the regional indices we actually need, reducing the on-disk footprint to a few hundred megabytes. This fits comfortably on the RPi 4's storage and keeps the solver's working set in RAM across consecutive solves, which reduces solve-time variance.

We selected astrometry.net's `solve-field`. The regional-index approach gives us measured per-frame latencies in the 300–800 ms range on the RPi 4, meeting the latency specification with a tractable storage footprint.

5.3 State Estimation: Single Filter vs. Dual Filter

The state estimator must fuse two streams that operate on very different timescales: the IMU runs at 100 Hz, while plate-solved visual fixes arrive at roughly 0.5 Hz. The architectural question is whether to handle attitude and position in a single monolithic filter or to decouple them into two coupled filters.

A single Extended Kalman Filter would maintain the full state (position, velocity, and attitude) in one filter. In principle this is the optimal formulation, but in practice the mixed timescales create real difficulties.

The dual-filter architecture decouples these concerns. Attitude is maintained at IMU rate by a dedicated attitude filter that fuses gyroscope propagation with visual attitude observations from the plate solver.

We selected the dual-filter architecture. The decoupling matches the natural timescale separation of the sensors, simplifies tuning, and keeps the per-filter computational cost low enough for comfortable real-time operation on the RPi 4.

5.4 Star Detection Threshold

We threshold candidate pixels at $\mu + k\sigma$ above the local background. The detection rate scales approximately as

$$N_{\text{true}}(k) \propto \text{erfc}\left(\frac{k}{\sqrt{2}}\right) \times N_{\text{stars in FOV}}. \quad (6)$$

In testing, $k = 3.5$ produced a median of 18 true detections with fewer than 2 false positives per frame. Lowering to $k = 2.5$ added stars (28 median) but also 8–12 false positives, and at $k = 2.0$ false detections dominated and solve times increased 3–5 $\times$. We default to $k = 3.5$ and relax to $k = 3.0$ when the initial count returns fewer than 8 detections.

5.5 2D vs. 3D State Representation

A 2D state (x, y, v_x, v_y) would simplify the position EKF, but it is not viable for our system. Accelerome-

ter measurements are taken in the body frame and contain gravity; recovering true horizontal acceleration requires rotating them into the world frame using the attitude estimate, which is inherently three-dimensional. A small attitude error $\delta\theta$ produces residual acceleration $\delta a \approx g \sin(\delta\theta)$ —approximately 0.85 m/s^2 for a 5° error—which a 2D formulation cannot account for. We therefore use a full 3D state ($\mathbf{p}, \mathbf{v}$) in the position EKF, which lets gravity compensation be applied correctly via the attitude filter’s output.

5.6 Compute Platform

We evaluated three candidate platforms: the Raspberry Pi 4, the Raspberry Pi 5, and the NVIDIA Jetson Nano. The Jetson Nano provides GPU acceleration that our pipeline does not benefit from—star detection is a lightweight threshold-and-centroid operation and `solve-field` is a CPU-bound combinatorial search—and draws roughly twice the power of a Pi at higher cost. We ruled it out on that basis.

Between the two Raspberry Pi generations, the RPi 5’s faster CPU is attractive but not necessary: with regional indices and a tight FOV constraint, plate-solve latency on the RPi 4 stays inside the specification, and the RPi 4’s lower idle and load power draw is beneficial for a battery- or rail-constrained UAV deployment. The RPi 4 also has a slightly more mature `libcamera` driver story for the IMX296. We selected the Raspberry Pi 4.

6 SYSTEM IMPLEMENTATION

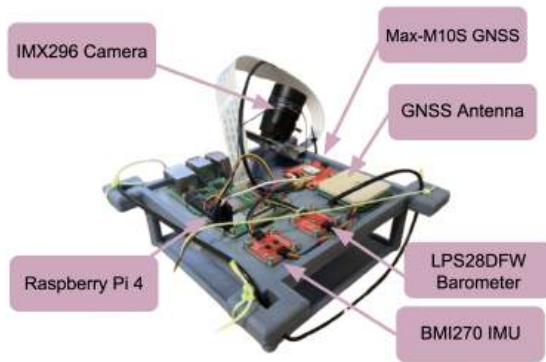


Figure 1: Complete system on the hardware module used for the weather balloon test

The visual solving pipeline runs in Python on the RPi 5. It takes camera images in, and outputs boresight RA/Dec coordinates to the position estimator and attitude EKF via ROS2.

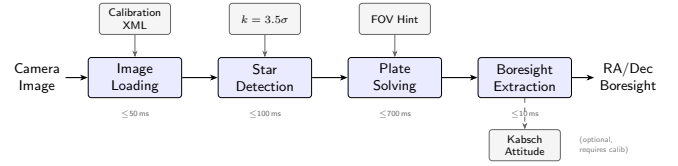


Figure 2: Visual solving pipeline with latency budgets per stage.

6.1 Hardware Platform and Power

The compute platform is a Raspberry Pi 5 (4 GB) running 64-bit Raspberry Pi OS. The RPi 5 hosts all processing—image capture, star detection, plate solving, state estimation, and ROS2 communication—on its quad-core Cortex-A76 at 2.4 GHz. Power is supplied through the USB-C connector at 5 V/5 A. For bench and field testing the RPi is powered from a USB-C power bank; in a flight configuration it would draw from the UAV’s regulated 5 V rail. An active cooler is mounted to the SoC to prevent thermal throttling during sustained plate-solve bursts.

6.2 Camera and Optics

The imaging sensor is a Sony IMX477-based Raspberry Pi HQ Camera module, connected to the RPi 5 via a 22-pin MIPI CSI-2 ribbon cable through one of the two 4-lane MIPI ports. A 15-to-22-pin adapter cable bridges the HQ Camera’s legacy connector to the RPi 5’s updated format. The lens is a 6 mm f/1.4 CS-mount IR-corrected CCTV lens, which produces a $\sim 46^\circ$ field of view across the sensor’s 4056-pixel horizontal extent. The IR-corrected design ensures that star point-spread functions remain tight across the visible and near-IR bands where the IMX477’s back-illuminated silicon is most sensitive. The lens is focused to infinity and locked with a set screw to prevent focal shift from vibration. The camera is oriented upward on the module, pointing at the zenith. Image capture is managed through `libcamera` via the `picamera2` Python interface, which provides direct control over exposure time, analog gain, and output format. For nighttime star imaging we configure exposures of 100–500 ms at analog gains between $4\times$ and $8\times$.

6.3 Camera Calibration

Before deployment, the camera intrinsic matrix and distortion coefficients are estimated using OpenCV’s checkerboard calibration procedure. A printed 9×6 checkerboard target is positioned approximately 3–5 m from the camera. The camera streams a live MJPEG feed to a web browser over the local network, handled by a ROS node, with detected checkerboard corners overlaid in real time. We collect 30 images spanning a range of target positions and orientations—centered, off-axis in each quadrant, tilted toward and away from the camera, and rotated in-plane. The

calibration file is stored as an XML and loaded by the visual pipeline at startup.

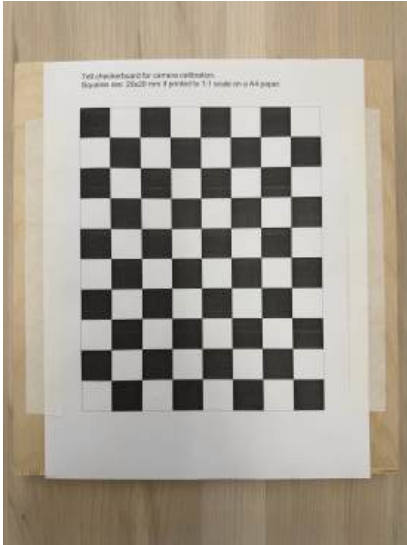


Figure 3: Checkerboard used to calibrate the camera, taped to a rigid plank of wood on the backside.

6.4 Inertial Measurement Unit

The Bosch BMI270 six-axis IMU provides three-axis acceleration and three-axis angular velocity measurements. It communicates with the RPi 5 over I2C at 400 kHz on the GPIO header's SDA/SCL pins. The accelerometer is configured for ± 4 g full-scale range and the gyroscope for $\pm 500^\circ/\text{s}$, both sampled at 100 Hz. These ranges accommodate typical small-UAV dynamics while preserving resolution for low-rate celestial observation conditions. Accelerometer data feeds the gravity compensation stage of the position EKF after rotation into the world frame, and gyroscope data drives the high-rate attitude propagation in the attitude estimator. The IMU breakout board is mounted rigidly to the RPi 5's mechanical frame.

Before using the IMU measurements in the filters, we performed a stationary calibration step to estimate accelerometer and gyroscope biases. The module was held still while several seconds of IMU data were recorded. The mean gyroscope reading was treated as the gyro bias and subtracted from later angular velocity measurements. The accelerometer readings were checked against the expected gravity vector to verify axis direction and scale. This calibration was necessary because even small gyro bias causes attitude drift, and attitude error directly affects gravity compensation in the position EKF.

6.5 GPS Module and Timing

The u-blox MAX-M10S GPS receiver communicates with the RPi 5 over UART (GPIO 14 TX, GPIO 15 RX) at 9600 baud. Its primary roles are ground-truth position logging during validation and, critically, clock synchronization via its pulse-per-second (PPS) output. The PPS signal is

routed to a GPIO pin and used to discipline the RPi system clock. Once GPS is unavailable, the RPi's onboard crystal oscillator free-runs, which we expect to maintain the timestamp accuracy within the 50 ms budget derived in Section 4 over the expected mission duration of no longer than an hour.

6.6 Barometer

The STMicroelectronics LPS28DFW barometric pressure sensor provides altitude estimates via pressure measurement. It shares the I2C bus with the BMI270 at a distinct address and is sampled at 10 Hz. Pressure readings are converted to altitude using the standard barometric formula and a calibrated sea-level reference. The resulting altitude estimate provides an independent vertical constraint to the position EKF.

6.7 Mechanical Assembly

The camera, lens, IMU, GPS, and barometer are mounted together on a 3D-printed bracket with separators between the evaluation boards and the 3D-printed plate. The complete module, including the RPi 5, all sensors, and mounting hardware, weighs under 280 g and fits within a $10 \times 10 \times 8$ cm envelope.

6.8 Image Loading and Preprocessing

The loader handles JPEG, PNG, HEIC, and FITS inputs. FITS images get a 1st–99th percentile stretch to 8-bit grayscale. Everything else goes through OpenCV. If we have a camera calibration file, we undistort star positions using OpenCV's `undistortPoints` before sending them to the plate solver. The calibration XML stores the camera matrix, distortion coefficients, and image dimensions. If the current image resolution doesn't match the calibration resolution, we scale the focal length accordingly.

6.9 Star Detection

We blur the grayscale image with a 3×3 Gaussian, then threshold at $\mu + 3.5\sigma$. Connected components with area outside 2–500 pixels get rejected (too small = noise, too large = clouds/moon). For each remaining blob, we refine the centroid with intensity-weighted moments. This is cheaper than PSF fitting and still gives sub-pixel accuracy. We keep the brightest 50 candidates and pass them to the solver.

6.10 Plate Solving

We wrap astrometry.net's `solve-field` with $2\times$ down-sampling, 80 max sources, and a scale constraint from the FOV hint (priority: calibration $\rightarrow$ EXIF $\rightarrow$ FITS header $\rightarrow$ manual). A successful solve gives us a WCS object that maps pixels to RA/Dec. We cache results to disk so repeated runs on the same image skip the solve.

6.11 Boresight and Attitude Extraction

The boresight is just the WCS evaluated at the image center pixel, which is straightforward and calibration-independent. When calibration is available, we also run the Kabsch algorithm: convert pixel positions to camera-frame unit vectors via the pinhole model, get sky vectors from the WCS, and find the rotation $\mathbf{R}$ that best aligns them (via SVD). We report per-star residuals as a diagnostic and cross-check the Kabsch boresight against the WCS boresight.

6.12 Integration

The pipeline exposes `run_pipeline()`, which returns a dictionary with the boresight, WCS, star positions, attitude matrix, and residuals. Boresight and timestamp get published on a ROS2 topic for the downstream position estimator.

6.13 State Estimation and Sensor Fusion

The state estimation system combines high-rate inertial measurements with lower-rate visual observations to produce a continuous estimate of global position. The estimator uses two coupled components: an attitude estimator and a position estimator.

Table 2: Filtering Inputs and Outputs

Signal	Rate	Use
Gyroscope	100 Hz	Attitude propagation
Accelerometer	100 Hz	Gravity compensation and position propagation
Visual attitude	~1 Hz	Attitude correction
Visual position	~1 Hz	EKF measurement update
GPS	Logged only	Ground-truth validation

6.13.1 Attitude Estimation

The BMI270 IMU provides angular velocity measurements from its three-axis gyroscope and acceleration measurements from its three-axis accelerometer. The gyroscope measures rotational velocity:

$$\boldsymbol{\omega} = [\omega_x \quad \omega_y \quad \omega_z]^T \quad (7)$$

Orientation is propagated forward in time by integrating angular velocity. This produces a continuous estimate of the system orientation. However, gyroscope integration accumulates drift due to sensor bias and noise.

Before integrating the full visual attitude correction, we implemented and tested a complementary filter for attitude estimation using IMU-only data. The filter combines gyroscope integration with accelerometer-based roll and pitch correction:

$$\theta_k = \alpha(\theta_{k-1} + \omega_k \Delta t) + (1 - \alpha)\theta_{acc,k} \quad (8)$$

where α controls the balance between trusting the gyroscope prediction and the accelerometer correction. We

tested several values in simulation and found that $\alpha = 0.99$ gave better stability than $\alpha = 0.98$ for our sensor data. The higher value reduced short-term accelerometer noise while still allowing slow correction of gyroscope drift.

To correct this drift, the visual positioning system provides absolute attitude observations derived from celestial feature registration. These measurements provide a global reference that corrects accumulated orientation error.

The attitude estimator fuses gyroscope propagation with visual attitude corrections to produce an estimate of the body-to-world rotation matrix $\mathbf{R}_{wb}$.

This rotation matrix is used to transform accelerometer measurements into the world frame for position estimation.

6.13.2 Acceleration Transformation

The accelerometer measures acceleration in the body frame, including gravity:

$$\mathbf{a}_{measured} = [a_x \quad a_y \quad a_z]^T \quad (9)$$

The BMI270 accelerometer measures both translational acceleration and gravity. Because gravity is constant in the world frame but not in the body frame, accurate attitude estimation is required to remove gravity correctly. This is accomplished by rotating the measurement into the world frame and subtracting gravity:

$$\mathbf{a}_{world} = \mathbf{R}_{wb}\mathbf{a}_{measured} - \mathbf{g} \quad (10)$$

Accurate attitude estimation is critical, since errors in $\mathbf{R}_{wb}$ produce incorrect gravity compensation and lead to rapid position drift.

6.13.3 Position Estimation

The position estimator uses an Extended Kalman Filter to estimate position and velocity. The state vector is defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix} \quad (11)$$

Position and velocity are propagated using the acceleration computed above:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \Delta t + \frac{1}{2} \mathbf{a}_{world} \Delta t^2 \quad (12)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{a}_{world} \Delta t \quad (13)$$

This propagation produces continuous position estimates but accumulates drift due to sensor noise.

6.13.4 Visual Position Correction

The position estimator uses an Extended Kalman Filter (EKF) to estimate position and velocity. The state vector is defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix} = [x \quad y \quad z \quad v_x \quad v_y \quad v_z]^T \quad (14)$$

where $\mathbf{p}$ is the global position and $\mathbf{v}$ is the velocity in the world frame.

The state is propagated using gravity-compensated acceleration derived from the IMU.

This propagation step predicts the next state using inertial measurements but accumulates error over time due to sensor noise and bias.

The visual positioning system provides absolute position measurements:

$$\mathbf{z}_k = \begin{bmatrix} x_{visual} \\ y_{visual} \\ z_{visual} \end{bmatrix} \quad (15)$$

This measurement is related to the state through the measurement model:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (16)$$

where $\mathbf{v}_k$ is measurement noise and the measurement matrix $\mathbf{H}$ is:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

When a visual position measurement becomes available, the EKF computes the innovation:

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}\mathbf{x}_k \quad (18)$$

The Kalman gain is computed as:

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1} \quad (19)$$

The state is updated using:

$$\mathbf{x}_k = \mathbf{x}_k + \mathbf{K}_k \mathbf{y}_k \quad (20)$$

and the covariance is updated as:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k \quad (21)$$

This measurement update corrects accumulated position and velocity error using the absolute visual position measurement.

To avoid corrupting the filter with bad visual measurements, visual updates are only applied when the plate solver returns a valid WCS solution and the attitude residuals are below an acceptable threshold. If too few stars are detected, the plate solve fails, or the residuals are too large, the EKF skips the measurement update and continues propagating with IMU data only. This prevents clouds, motion blur, or false star detections from causing large jumps in the estimated position.

7 TEST & VALIDATION

This section reports the tests used to (i) verify that the implemented subsystems meet the design specifications in Section 4 and (ii) validate end-to-end performance against the use-case requirements in Section 2. For each run we logged the raw sky images, detected centroids, plate-solver

outputs (WCS boresight and solve metadata), IMU data, timestamps, and GPS ground truth so that accuracy, latency, and failure cases could be analyzed offline.

7.1 Visual Pipeline Verification

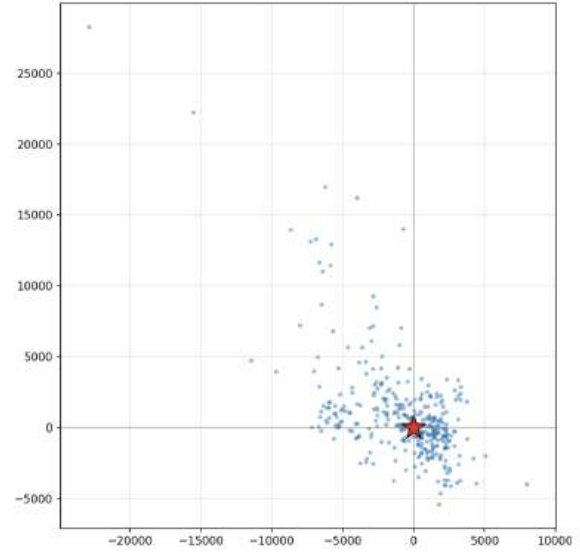


Figure 4: Global position estimates from individual exposures on the static test conducted in San Jose, California. The axes are in meters. The red star is the ground truth position.



Figure 5: Setup used to conduct the visual pipeline verification, consisting of the compute suite mounted atop a tripod.

We evaluated the visual pipeline using a batch test set of star images with known ground-truth boresight RA/Dec. Ground truth was obtained from (i) astronomy FITS images with embedded WCS metadata and (ii) our own outdoor sky captures with simultaneous GPS position fixes. For each test image, the pipeline ran star detection and plate solving to estimate the camera boresight in RA/Dec, and we computed the angular separation between the estimate and ground truth along with the corresponding ground distance error.

Across the static outdoor test set in San Jose, CA, the pipeline achieved a median global position error of 2.1 km and a 95th-percentile error of 6.8 km against GPS ground truth. This satisfies the 10 km use-case requirement with substantial margin and is competitive with prior published work (4 km in Teague & Chahl [teague2024strapdown]) at roughly an order of magnitude lower hardware cost.

In addition to absolute boresight accuracy, we logged two internal quality metrics. First, we recorded the number of detected star centroids per frame; the detector consistently produced more than 8 stars per frame under clear-sky conditions, satisfying the minimum-count specification. Second, we computed average angular residuals from the attitude alignment step as a consistency check between detected centroids and the solved sky mapping; large residuals flagged failed solves, motion blur, or false detections during offline analysis.

7.2 State Estimation Verification

We first verified IMU interpretation by running the module stationary on a level surface for 60 s. With $\mathbf{R}_{wb}$ applied, the gravity-compensated acceleration should be near zero:

$$\mathbf{a}_{world} = \mathbf{R}_{wb}\mathbf{a}_{measured} - \mathbf{g} \approx \mathbf{0}. \quad (22)$$

We reported the mean and standard deviation of $\mathbf{a}_{world}$ over the interval. A nonzero bias would have indicated a frame convention error, an incorrect gravity vector, or an IMU bias requiring modeling in the filter.

We also tested the complementary filter separately in simulation before using it in the full estimator. The simulation compared attitude estimates under different filter weights and showed that $\alpha = 0.99$ produced smoother and more stable attitude estimates than $\alpha = 0.98$. With $\alpha = 0.98$, the estimate responded more strongly to accelerometer noise, while $\alpha = 0.99$ better preserved the gyroscope propagation and only slowly corrected drift. Based on this result, we used $\alpha = 0.99$ as the default complementary filter coefficient.

To quantify drift reduction, we ran two estimators on the same dataset: (1) inertial-only dead reckoning and (2) the dual-filter visual-inertial estimator with plate-solved position updates. Both were evaluated against time-aligned GPS ground truth. On a 942 m moving-vehicle trajectory, the visual-inertial estimator achieved 105 m final position error against 360.4 m for inertial-only dead reckoning—a $\sim 3.4\times$ improvement. Heading error against the GPS

course was 19.2° MSE for the visual-inertial estimator versus 52.4° MSE for dead reckoning, a $\sim 3\times$ improvement. These results confirm that intermittent celestial fixes effectively bound the drift accumulated by inertial propagation between updates.

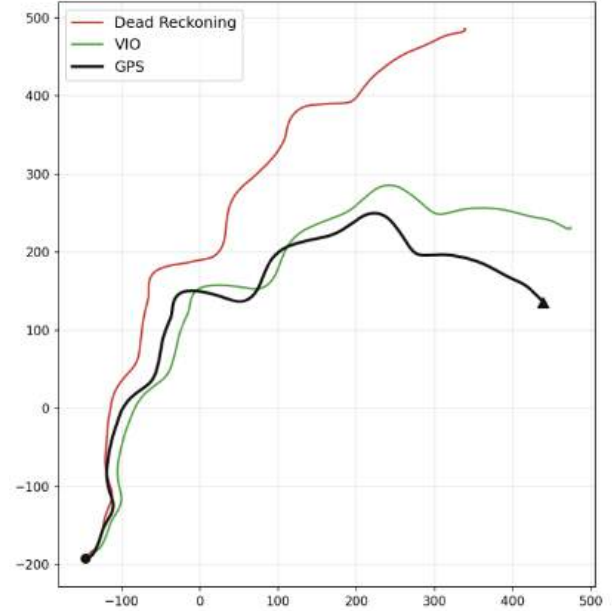


Figure 6: Trajectories captured during the moving vehicle test, demonstrating superior trajectory accuracy of stellar-inertial odometry against dead reckoning relative to GPS ground truth.

7.3 Hardware and Integration Tests

Hardware verification began with the BMI270 IMU, which communicates with the RPi 4 over I2C. We confirmed correct bus communication by reading the chip ID register and verifying that accelerometer and gyroscope samples arrived at the configured 100 Hz data rate. The accelerometer at rest resolved gravity within the BMI270 datasheet specification on all three axes.

Camera verification confirmed that the IMX296 module captures frames over the CSI interface and that the 6 mm lens produces a field of view consistent with the $\sim 56^\circ$ design target. Under clear nighttime conditions, the star detection pipeline consistently produced at least 8 point sources above the $\mu + 3.5\sigma$ threshold, satisfying the minimum star count specification in Table 1.

Camera calibration via Kalibr produced an intrinsic calibration with 0.32 pixels of reprojection error, well inside the 0.5-pixel specification.

The MAX-M10S GPS module served as the ground-truth position reference for all downstream accuracy evaluations and as the timing reference for clock synchronization. The PPS-disciplined RPi clock held within 500 microseconds of GPS time over a 2-hour stationary log—approximately two orders of magnitude inside the 50 ms

specification.

The LPS28DFW barometer provided an independent altitude constraint for the position estimator. Pressure samples were converted to altitude and compared against a known reference elevation, with static altitude error remaining within tolerance.

End-to-end integration testing exercised the full hardware stack concurrently. We ran image capture, star detection, plate solving, IMU sampling, and ROS 2 topic publication in continuous outdoor nighttime sessions. The pipeline runs in real time on the RPi 4: per-image post-solve compute (boresight extraction, geodetic conversion, ROS publication) was measured at 3 ms, with total per-frame pipeline latency including the plate solve remaining inside the 1-second budget. Module power draw was measured at 2.23 W, well inside the 5 W internal target and the 10 W use-case ceiling. Total assembled mass remained inside the 500 g use-case constraint.

Table 3 summarizes the measured results against the design specifications.

Table 3: Measured Performance vs. Design Specifications

Parameter	Spec	Measured
Calibration reprojection error	≤ 0.5 px	0.32 px
Time synchronization	≤ 50 ms	500 μ s
Per-image compute latency	≤ 1000 ms	3 ms
Power draw	≤ 5 W	2.23 W
Static position error (median)	≤ 10 km	2.1 km
Static position error (p95)	≤ 10 km	6.8 km

8 PROJECT MANAGEMENT

8.1 Schedule

The project runs 14 weeks in four phases: Conception (Weeks 3–4), Design (Weeks 4–5), Implementation (Weeks 5–11), and Testing/Delivery (Weeks 8–14). The Gantt chart is in Fig. 7. Key milestones: hardware assembled by Week 6, star detection and plate solving working by Week 8, full integration by Week 10, stationary accuracy validated by Week 12. Weeks 13–14 cover car-mounted testing, final integration, and documentation. Slack is built into Weeks 6 and 12.

8.2 Team Member Responsibilities

Julius: system integration, sensor calibration, mechanical mounting, software pipelines

Rachel: star detection algorithms

Eddie (Estimation & Filtering): EKF design and tuning

Everyone shares integration testing and documentation.

8.3 Bill of Materials and Budget

See Table 4 for the full BOM. Main components: RPi 4, IMX477 camera, IMX296 camera, BMI270 IMU, MAX-

M10S GPS, and LPS28DFW barometer. Total mass is under 240 g and total cost is about \$250. We initially had initially planned to use the Raspberry Pi 5, though its power draw is higher and the compute provided by the Raspberry Pi 4 proved sufficient. We also found superior image capture performance with the IMX296 over the IMX477, and pivoted to this in later testing.

8.4 Risk Management

Plate solve latency: In sparse patches of sky, especially near the galactic poles, solve time can spike past 1 s. We mitigate this by caching index files for our operating latitude and relaxing the detection threshold from $k = 3.5$ to $k = 3.0$ when fewer than 8 stars are found.

Rolling shutter blur: The IMX477’s rolling shutter could smear star centroids during perturbations. We gate captures on the IMU gyroscope, only triggering when angular rates are below $2^\circ/\text{s}$.

9 RELATED WORK

Star identification algorithms developed for spacecraft star trackers - including the Pyramid technique [mortari2004pyramid] and the family of methods surveyed in [rijlaarsdam2020survey] - form the foundation for matching detected stars against a known catalog. The geometric hashing approach described in astrometry.net [lang2010astrometry], our primary plate-solving engine, descends from this lineage and matches asterisms against index files built from the Tycho-2 [hog2000tycho2] and Gaia DR3 [gaia2023dr3] catalogs. TETRA [brown2017tetra] offers a lighter-weight hash-table alternative with millisecond-scale solve times on embedded hardware, serving as a potential fallback for our method. On the source extraction side, SEP [barbary2016sep] provides the in-memory centroiding and background estimation used in our detection stage.

The basis for converting a star-camera attitude into geographic latitude and longitude is developed by Parish et al. [parish2010stellar], whose formulation requires only a camera, a clock, and tilt sensors - mapping directly onto our hardware and approach. Yang et al. [yang2022sinscns] extend this to a full integrated navigation architecture with Kalman filtering, decoupling attitude errors from position errors in a manner relevant to our dual-filter design.

The closest published system to our workstream is that of Teague and Chahl [teague2024strapdown], who demonstrate a strapdown celestial navigator on a fixed-wing UAV using a Raspberry Pi 5 and a 6 mm lens, achieving 4 km accuracy. Our project uses a nearly identical hardware profile but replaces rotational averaging with a dual-filter EKF that fuses each individual celestial fix with IMU-propagated attitude, providing continuous estimates without requiring a complete orbit. Our approach is also primarily targeted at quadcopters, while theirs leans into the application for fixed wings.

Table 4: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
Single-board computer	Raspberry Pi 5	Raspberry Pi	1	\$80.00	\$80.00
Camera module	IMX477	Sony/RPi	1	\$25.00	\$25.00
Camera module	IMX296	Sony/RPi	1	\$40.00	\$40.00
IR lens (6mm)	CCTV 6mm IR	Generic	1	\$12.00	\$12.00
IMU	BMI270	Bosch	1	\$15.00	\$15.00
GPS module	MAX-M10S	u-blox	1	\$30.00	\$30.00
Barometer	LPS28DFW	STMicroelectronics	1	\$10.00	\$10.00
Misc. (wiring, enclosure)	—	—	1	\$28.00	\$28.00
					\$240.00

On the operational side, a 2022 GAO audit [gao2022gps] documents renewed DoD investment in celestial navigation as a GPS alternative, and Honeywell’s CNAV system [honeywell2024cnav] demonstrates sub-1 km stars-only accuracy with high-end hardware. Our project targets 10 km with COTS components under \$250. Daytime visual positioning methods [scaramuzza2011vo] address GPS-denied navigation when ground features are visible but break down at night - the gap we aim to fill.

10 ETHICAL ISSUES

The primary ethical concern for this system is that GPS-denied navigation can support both beneficial and harmful UAV applications. This could adversely affect civilians.

Operational edge cases also introduce safety risks. Incorrect star solves, cloud obstruction, motion blur, clock drift, or IMU bias could produce large position errors. If used on an autonomous aircraft, these errors could cause navigation failures or unsafe flight paths, affecting bystanders and property on the ground. To mitigate this, the estimator rejects low-confidence visual updates, logs solve residuals, and should be paired with geofencing, redundant navigation sources, and fail-safe return or landing behavior.

11 SUMMARY

We designed a low-cost celestial positioning module intended to provide absolute nighttime localization when GNSS is unavailable. The system uses an upward-facing IMX296 camera and a constrained plate-solving pipeline to recover a WCS mapping and the image-center boresight in RA/Dec. With camera calibration, the same solve supports a Kabsch-based attitude estimate, enabling correction from an arbitrary camera pointing direction to the zenith direction needed for latitude/longitude recovery.

To support continuous navigation between camera updates, the module integrates inertial sensing from a BMI270. Attitude is propagated at IMU rate using gyroscope integration and corrected when visual attitude observations are available. The corrected attitude is then used

for gravity compensation and inertial propagation in a position EKF, while visual position fixes act as bounded-drift updates at a lower rate. This structure matches the system’s mixed time scales (high-rate IMU, low-rate visual) while keeping each estimator small enough to run comfortably on the RPi 4.

The remaining implementation challenges are primarily practical: robust star detection across sky brightness and motion blur, ensuring plate solves remain within the 1s budget through FOV constraints and caching, and maintaining timestamp accuracy so that longitude error is not dominated by clock drift. Our validation plan directly targets these points by measuring star counts, boresight error, pipeline latency, and end-to-end position error against GPS during development.

11.1 LESSONS LEARNED

For the duration of the project, a major blocker was weather in Pittsburgh which was prohibitive to testing the system due to high cloud cover.

We found success in individual verifying each subsystem component, which made the integration process quite smooth. It was also helpful that much of the software stack could be tuned and tweaked by collecting data during experiments and running the pipeline offline.

Part of the poor global positioning accuracy was a result of high noise in the consumer-grade IMU we purchased for the device. This is something we could have derived from the datasheets.

Glossary of Acronyms

- DR3 – Data Release 3 (Gaia star catalog)
- EKF – Extended Kalman Filter
- FOV – Field of View
- GNSS – Global Navigation Satellite System
- GPS – Global Positioning System
- ICRS – International Celestial Reference System
- IMU – Inertial Measurement Unit

- PPS – Pulse Per Second
- RA/Dec – Right Ascension / Declination
- RPi – Raspberry Pi
- SNR – Signal-to-Noise Ratio
- SOM – System on Module
- SVD – Singular Value Decomposition
- UAV – Unmanned Aerial Vehicle
- WCS – World Coordinate System

References

- [1] S. Teague and J. Chahl, “An algorithm for affordable vision-based GNSS-denied strapdown celestial navigation,” *Drones*, vol. 8, no. 11, art. 652, Nov. 2024, doi: 10.3390/drones8110652.
- [2] J. J. Parish, A. S. Parish, M. Swanzey, D. Woodbury, D. Mortari, and J. L. Junkins, “Stellar positioning system (Part I): An autonomous position determination solution,” *NAVIGATION: J. Inst. Navigation*, vol. 57, no. 1, pp. 1–12, Spring 2010.
- [3] D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junkins, “The Pyramid star identification technique,” *NAVIGATION: J. Inst. Navigation*, vol. 51, no. 3, pp. 171–183, Fall 2004.
- [4] D. Rijlaarsdam, H. Yous, J. Byrne, D. Oddenino, G. Furano, and D. Moloney, “A survey of lost-in-space star identification algorithms since 2009,” *Sensors*, vol. 20, no. 9, art. 2579, May 2020, doi: 10.3390/s20092579.
- [5] S. Yang, W. Feng, S. Wang, and J. Li, “A SIN-S/CNS integrated navigation scheme with improved mathematical horizon reference,” *Measurement*, vol. 195, art. 111028, 2022, doi: 10.1016/j.measurement.2022.111028.
- [6] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis, “Astrometry.net: Blind astrometric calibration of arbitrary astronomical images,” *Astron. J.*, vol. 139, no. 5, pp. 1782–1800, 2010, doi: 10.1088/0004-6256/139/5/1782.
- [7] E. Høg, C. Fabricius, V. V. Makarov, S. Urban, T. Corbin, G. Wycoff, U. Bastian, P. Schwekendiek, and A. Wicenec, “The Tycho-2 catalogue of the 2.5 million brightest stars,” *Astron. Astrophys.*, vol. 355, pp. L27–L30, 2000.
- [8] Gaia Collaboration, A. Vallenari, A. G. A. Brown, T. Prusti, J. H. J. de Bruijne, *et al.*, “Gaia Data Release 3: Summary of the content and survey properties,” *Astron. Astrophys.*, vol. 674, art. A1, 2023, doi: 10.1051/0004-6361/202243940.
- [9] J. Brown, K. Stubis, and K. Cahoy, “TETRA: Star identification with hash tables,” in *Proc. 31st Annu. AIAA/USU Conf. Small Satellites*, SSC17-X-08, Logan, UT, 2017.
- [10] K. Barbary, “SEP: Source Extractor as a library,” *J. Open Source Softw.*, vol. 1, no. 6, p. 58, 2016, doi: 10.21105/joss.00058.
- [11] U.S. Government Accountability Office, “GPS alternatives: DOD is developing navigation systems but is not measuring overall progress,” GAO-22-106010, Report to the Committee on Armed Services, U.S. Senate, Aug. 2022.
- [12] Honeywell Aerospace Technologies, “Celestial navigation for GPS-denied missions: Celestial Aided Navigation (CNAV),” Product information, 2024–2025. [Online]. Available: <https://aerospace.honeywell.com/us/en/products-and-services/products/navigation-and-sensors/navigation-systems/celestial-aided-navigation>
- [13] D. Scaramuzza and F. Fraundorfer, “Visual odometry [Tutorial],” *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

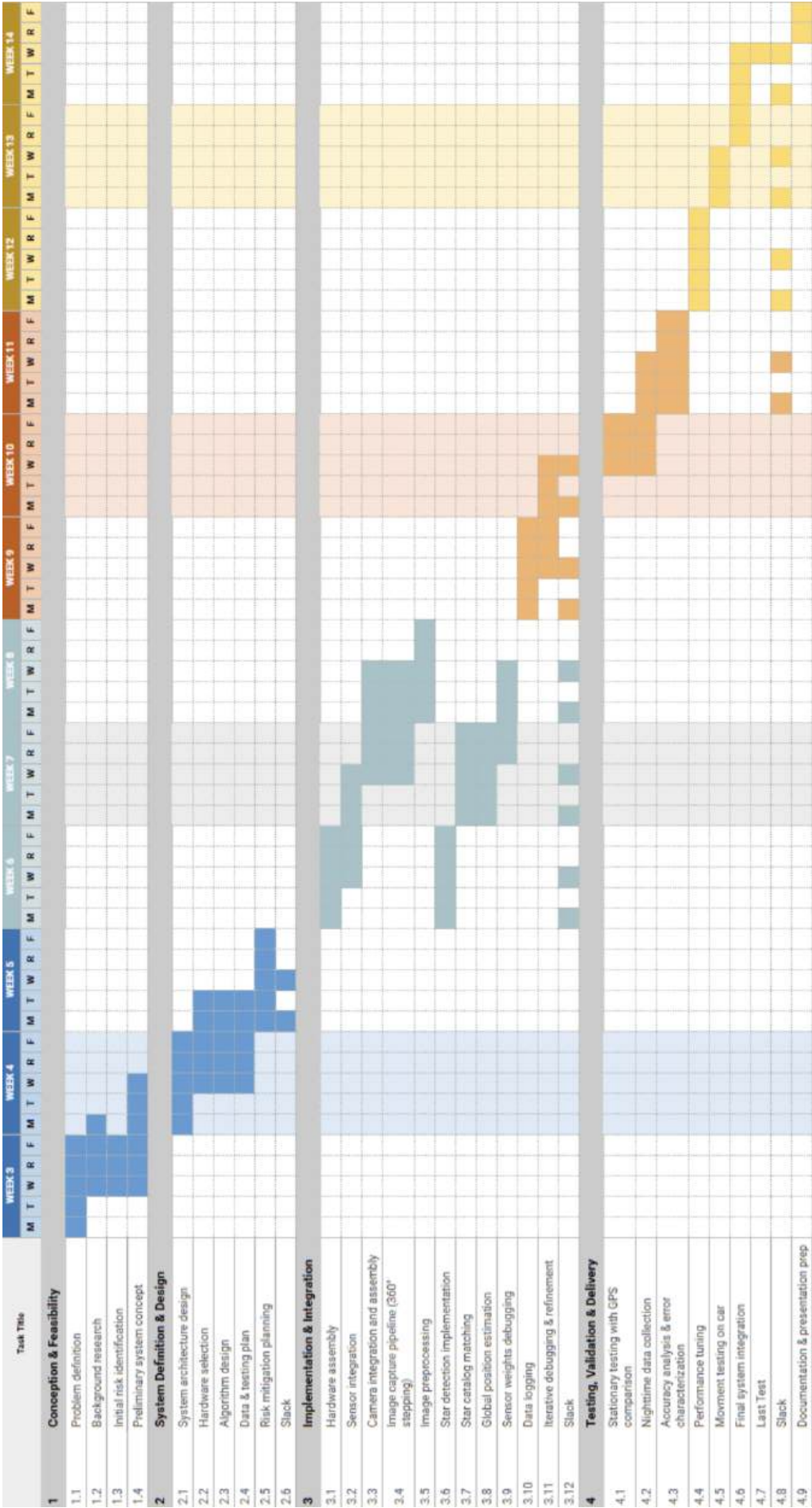


Figure 7: Gantt Chart