Project Proposal

Visual-Inertial Navigation for Autonomous UAV Operation in GPS-Restricted Environments

Project Supervisors - Internal:

Prof. Chandana Gamage

Group Members:

D. V. A. I. Delgahagoda	210113L
T. T. Kashmeera	210279A
K. D. R. P. Rathnayake	210537N

Department of Computer Science and Engineering
University of Moratuwa

Table of Content

Table of Figures

1 Introduction

1.1 Overview of UAVs and Their Applications

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, have significantly transformed various sectors, ranging from agriculture to logistics [1]. Their versatility and capacity to access remote or hazardous areas have positioned them as invaluable tools across numerous applications [2]. These applications encompass aerial mapping, surveillance, and inspection, as well as autonomous flight operations, facilitating data collection in environments otherwise inaccessible to human operators. The expanding utility of UAVs in diverse and complex settings underscores the increasing demand for advanced navigational capabilities.

1.2 The Critical Role of Navigation Systems

Autonomous UAV operation is fundamentally dependent on accurate and robust navigation and localization capabilities. Precise estimations of a UAV's position and orientation are indispensable for executing critical tasks such as obstacle avoidance, sophisticated path planning, and maintaining flight stability throughout a mission. Without reliable navigation, the potential of autonomous drones remains severely constrained, limiting their deployment to less complex, open environments [3].

1.3 Challenges in GPS-Restricted Environments

While the Global Positioning System (GPS) serves as a primary navigation tool for UAVs in many scenarios, its reliability diminishes significantly or becomes entirely unavailable in certain challenging environments. These challenging areas include dense urban landscapes, often termed "urban canyons," indoor facilities, under-canopy forest environments, and remote regions characterized by obstructed satellite visibility [4], [5]. The expansion of UAV applications into these previously inaccessible or difficult environments directly highlights the limitations of GPS, making the development of alternative or complementary navigation solutions imperative. This evolution of UAV capabilities necessitates more sophisticated and robust navigation systems that are less reliant on external signals and can leverage onboard sensor data.

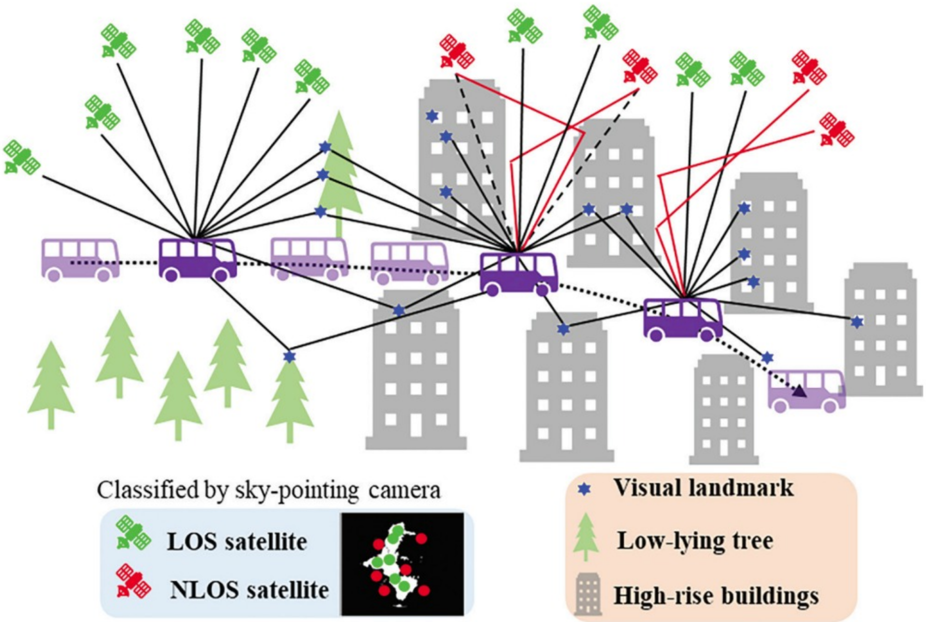


Figure 1: GNSS signal classification with LOS/NLOS satellites and visual landmarks in urban environments [6].

1.4 Introduction to Visual-Inertial Navigation Systems (VINS)

Visual-Inertial Navigation Systems (VINS) have emerged as a potent solution to overcome the limitations inherent in GPS-dependent navigation. VINS integrates visual information captured by onboard cameras with motion data provided by Inertial Measurement Units (IMUs). This fusion allows VINS to leverage computer vision algorithms for feature extraction and tracking, combined with advanced sensor fusion techniques, to accurately estimate the UAV's position, velocity, and orientation, even in the complete absence of GPS signals. The ability of VINS to provide precise, real-time pose estimation in challenging conditions positions it as a foundational technology for next-generation UAV autonomy. This means VINS is crucial for enabling UAVs to perform complex tasks, avoid obstacles, and maintain safety in dynamic and unstructured environments, pushing the boundaries of what autonomous drones can achieve [7], [8].

1.5 Purpose and Scope of the Project

This project proposes the implementation of a robust VINS specifically engineered for autonomous UAV operation in GPS-restricted environments. The primary aim is to significantly enhance navigation accuracy, reliability, and safety in such challenging scenarios. The project's scope encompasses a comprehensive analysis of existing VINS technologies, localization technologies, the identification of key challenges, and the subsequent development of a methodology that incorporates advanced techniques to overcome these identified limitations.

2 Problem Statement

2.1 Limitations of GPS in Challenging Environments

The efficacy of Global Navigation Satellite Systems (GNSS), particularly GPS, is severely hampered in environments that present physical obstructions or signal interference. GPS signals are inherently weak or entirely absent in indoor spaces, underground areas, and remote locations, rendering traditional satellite-based navigation ineffective.

In dense urban environments, commonly referred to as urban canyons, tall buildings obstruct the direct line-of-sight (LoS) to satellites. This leads to pervasive signal blockages and severe multipath propagation, where signals bounce off structures, distorting their accuracy. Such conditions cause a drastic degradation in positioning accuracy, potentially dropping from several meters to hundreds of meters [4], [5]. Similarly, under-canopy forest environments present dense obstacles and significant GNSS signal interference, making autonomous navigation for data collection exceptionally challenging.

Beyond direct signal issues, these environments often exhibit varying lighting conditions, low texture, or the presence of dynamic objects. These factors further complicate visual navigation components that might otherwise compensate for GPS loss [9]. The problem is not merely a singular environmental factor, but a synergistic combination of factors. For instance, urban canyons induce both signal blockage and multipath effects, while simultaneously presenting visually complex scenes with dynamic elements. This multi-faceted degradation creates a compounding effect on navigation challenges, necessitating solutions that address multiple, interconnected environmental challenges simultaneously.

2.2 Impact on Autonomous UAV Operations

The limitations of GPS in challenging environments have profound implications for autonomous UAV operations. The primary challenge lies in maintaining precise positioning and navigation capabilities when satellite-based systems become unreliable or completely unavailable. Inaccurate localization leads to significant deviations from planned flight paths, as evidenced by experiments where drones deviated by 1 to 5 meters or more in urban settings [10].

Such navigation disruptions affect both autonomous and manual flight, particularly during critical phases like landing and obstacle avoidance. In loosely coupled navigation systems, a complete data outage can occur when fewer than four satellites are visible, leading to a loss of positional awareness. Unreliable navigation consequently jeopardizes the safety of UAV missions, especially in complex or dynamic environments where collision avoidance and precise task execution are paramount.

This inability to operate reliably in GPS-restricted areas severely constrains the potential applications and mission capabilities of autonomous UAVs. This highlights a "last mile" problem for UAV autonomy. While long-range navigation in open areas might be feasible with GPS, the most complex and often most valuable operations, such as precise delivery in a city, detailed indoor inspection, or surveying under dense foliage, occur precisely where GPS is least reliable. Solving this "last mile" navigation problem with robust VINS is crucial for unlocking the full potential and economic value of autonomous UAVs in high-impact, real-world scenarios, moving beyond simple line-of-sight operations.

3 Research Objectives

Visual-Inertial Navigation Systems (VINS) offer a powerful framework for enabling UAV autonomy in GPS-restricted environments. However, their practical deployment is challenged by several limitations that arise from sensor imperfections, environmental conditions, and computational constraints. To ensure reliable navigation and robust performance, it is essential to explicitly address these key issues. The main problems this research seeks to overcome are outlined below:

3.1 Inertial-Only IMU Drift

Inertial-only navigation (using only IMU data) accumulates sensor errors over time, causing the estimated trajectory to drift unboundedly [10]. In practice, biases and noise integrate into large position/attitude errors if not corrected by external references. Without GPS or vision updates, this long-term drift makes pure INS unreliable for GPS-denied flight.

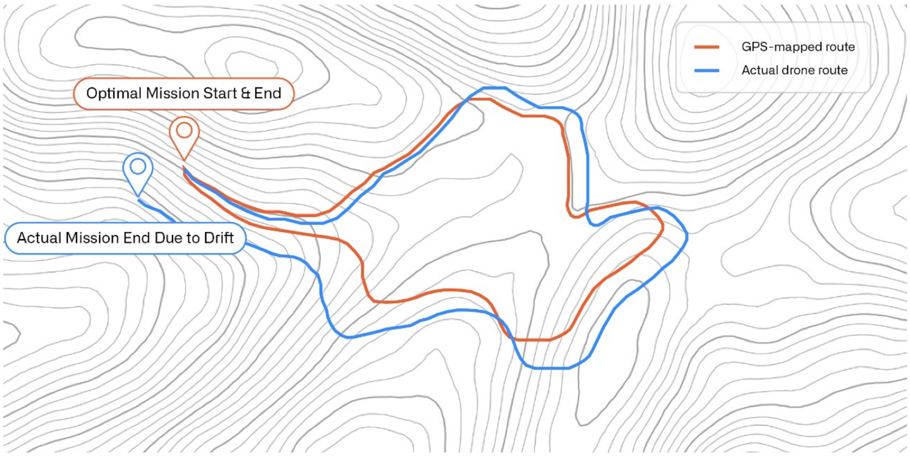


Figure 2: Visualization of IMU drift in UAVs [11].

3.2 Visual Tracking in Low-Texture/Dynamic Scenes

Vision-based navigation relies on tracking environmental features. In low-texture or uniform areas (e.g., blank walls or repetitive patterns), feature descriptors lack distinctiveness and matching often fails [12]. Likewise, highly dynamic scenes with many moving objects leave few stable landmarks, causing the visual tracker (and thus VINS) to lose lock or fail completely [12].

3.3 VINS Initialization and Scale Ambiguity

Monocular VINS cannot infer absolute scale from vision alone, so the initialized map's scale is ambiguous [13]. Proper initialization also requires rich motion (translations/rotations) to excite all degrees of freedom; for example, VINS-Mono must execute full 3-axis maneuvers to observe gravity and scale, which is time-consuming and resource-intensive [10]. If these conditions are not met, the estimator may diverge or converge to a wrong scale.

3.4 Real-Time VINS Computational Demands

Tightly-coupled VINS algorithms fuse high-rate IMU and image data via nonlinear optimization or filtering, which is computationally expensive. For instance, VINS-Mono's sliding-window bundle-adjustment ran at only ~9 Hz on typical UAV hardware, failing to meet real-time requirements on a low-cost platform [10]. In general, ensuring real-time performance requires significant CPU/GPU resources, which may be beyond small UAV capabilities.

3.5 Compound Environmental Degradation (Poor Lighting, Motion Blur, Dynamic Objects)

In severely degraded conditions combining multiple factors, VINS performance collapses. Extensive motion blur (from fast motion or low light) prevents reliable feature tracking [14], and scenes dominated by dynamic objects (e.g., >80% moving targets) provide too few static cues, leading to localization failure [13]. Together, these adverse effects can overwhelm the system, causing the estimator to lose track and diverge.

4 Literature Review

4.1 The Challenge of Autonomous Navigation in GPS-Restricted Environments

4.1.1 The Imperative for GPS-Independent Navigation

The proliferation of Unmanned Aerial Vehicles (UAVs) has catalyzed transformative advancements across a multitude of sectors, including surveillance, reconnaissance, infrastructure inspection, and search-and-rescue operations.[15] The operational envelope of these autonomous systems is increasingly expanding into complex and unstructured environments where the reliance on traditional navigation aids, primarily the Global Navigation Satellite System (GNSS), is untenable. Environments such as dense urban canyons, subterranean spaces, cluttered indoor settings, and thick forests are characterized as GPS-restricted or GPS-denied.[16] Within these domains, GNSS signals are frequently attenuated, blocked, or corrupted by multipath effects, where signals reflect off surfaces before reaching the receiver, leading to significant positional errors.[16] This inherent unreliability is further compounded by the growing threat of deliberate signal interference. Malicious actors can employ jamming techniques to overwhelm GNSS receivers with noise, or more insidiously, spoofing techniques to broadcast false signals, deceiving the UAV about its true location.[16] The strategic and commercial implications of this vulnerability are significant; recent reports indicate a sharp rise in GPS jamming and spoofing incidents, with as many as 700 events occurring globally each day, particularly in conflict zones and across North America and Europe.[17] The operational integrity of a UAV hinges on its ability to perform high-accuracy, continuous state estimation—the process of determining its position, velocity, and orientation (attitude) in real-time. The potential for GNSS failure, whether environmental or adversarial, renders it an insufficient standalone solution for robust autonomous flight, creating a critical demand for alternative navigation technologies.[18]

4.1.2 Visual-Inertial Navigation Systems (VINS) as a Premier Solution

Among the suite of alternative navigation technologies, Visual-Inertial Navigation Systems (VINS) have emerged as a premier solution, offering a compelling balance of performance, cost, and physical footprint. VINS achieves robust state estimation by fusing data from two complementary, low-cost, and lightweight sensors: a visual camera and an Inertial Measurement Unit (IMU).[19] These sensors are ubiquitous on modern robotic platforms and possess symbiotic characteristics. The camera, a passive sensor, captures rich information about the surrounding environment, including visual features such as color and texture.[16] Using computer vision algorithms, a VINS can identify and track these features across consecutive images to infer its own motion and, in many cases, build a map of its surroundings—a process known as Visual SLAM (VSLAM).1 However, vision-only systems are susceptible to failure in visually degraded conditions, such as low light, textureless environments (e.g., white walls), or during rapid motion that induces significant image blur.[20] The IMU, an active sensor, provides high-frequency measurements of the UAV's linear acceleration and angular velocity.[19] These measurements are self-contained and immune to external environmental conditions, allowing the system to continue estimating its motion even when visual information is unavailable.[20] The fundamental limitation of an IMU, particularly the low-cost Micro-Electro-Mechanical Systems (MEMS) type used in UAVs, is that its measurements are corrupted by noise and biases. When these measurements are integrated over time to compute velocity and position, the errors accumulate rapidly, leading to unbounded drift.[21] VINS masterfully exploits the complementary nature of these two sensors.[20] The high-frequency inertial data from the IMU effectively bridges the gaps between camera frames and provides robustness against visual degradation.[22] Concurrently, the visual data, which provides drift-free measurements relative to the static environment, is used to continuously correct for the IMU's accumulating drift.8 This tight integration allows for the real-time estimation of the vehicle's full six-Degrees-of-Freedom (6-DOF) state (position and orientation), which is the core problem that VINS aims to solve.[18]

4.1.3 Scope and Structure of the Review

This literature review provides a comprehensive analysis of Visual-Inertial Navigation Systems tailored for autonomous UAV operation in GPS-restricted environments. The review focuses primarily on the fundamental principles underpinning VINS and the seminal algorithms that have defined the state-of-the-art. Section 2 deconstructs the foundational sensor components, the IMU and the camera, detailing their operating principles and critical error models. Section 3 explores the core architectural paradigms that govern

VINS design, comparing data fusion strategies (loosely- vs. tightly-coupled) and back-end state estimation methodologies (filtering vs. optimization). Section 4 presents a detailed review of four prominent and widely-used VINS algorithms: MSCKF, OKVIS, VINS-Mono, and ORB-SLAM3, analyzing their key innovations, strengths, and limitations. Finally, Section 5 discusses the unique challenges of implementing VINS on UAV platforms and examines emerging trends and future research directions that are shaping the next generation of autonomous navigation systems.

4.2 Foundational Components of Visual-Inertial Systems

A deep understanding of VINS requires a thorough analysis of its constituent sensors. The performance of any VINS algorithm is fundamentally limited by the quality of the data it receives and the fidelity of the models used to interpret that data. This section details the working principles and, critically, the error characteristics of the IMU and the visual camera, which are essential for designing high-performance estimation algorithms.

4.2.1 The Inertial Measurement Unit (IMU): Principles and Error Modeling

Working Principles of MEMS IMUs An Inertial Measurement Unit is an electromechanical device that measures a body's specific force and angular rate using a combination of accelerometers and gyroscopes.[21] For UAV applications, MEMS-based IMUs are the standard choice due to their exceptional advantages in size, weight, power, and cost (SWaP-C).[21]

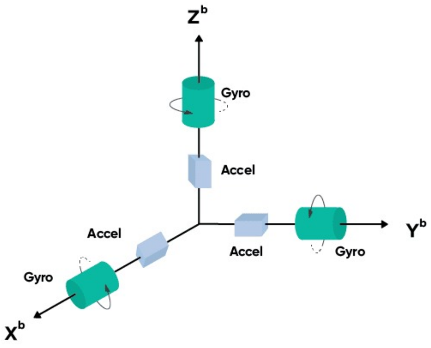


Figure 3: Image depicting the accelerometers and gyroscopes in the three axes of movement. Each accelerometer and gyroscope is positioned at 90° to the others (orthogonally). Accelerometers measure motion along each axis and each gyroscope measures angular velocity around each axis.[21]

A typical MEMS accelerometer operates on the principle of a proof mass suspended by springs within a reference frame. When the unit accelerates, the inertia of the proof mass causes it to displace relative to the frame. This displacement is measured, often by detecting the change in capacitance between the proof mass and a set of fixed electrodes, and is directly proportional to the applied linear acceleration.[21] A MEMS gyroscope commonly utilizes the Coriolis effect to measure angular velocity. A proof mass is driven to resonate along a specific axis. When the entire unit is subjected to rotation, the Coriolis force induces a secondary vibration in the proof mass along an axis perpendicular to both the drive axis and the axis of rotation. The magnitude of this secondary vibration, also measured capacitively, is proportional to the angular rate of the gyroscope.[21]

The IMU Measurement Model and Error Formulation The raw outputs from a low-cost MEMS IMU are not a direct measure of the true kinematics; they are corrupted by a combination of deterministic and stochastic errors that must be accurately modeled to prevent rapid degradation of the navigation solution.[23] The standard measurement model for a gyroscope and an accelerometer includes several key error terms:

- **Bias (b_g, b_a):** This represents a persistent offset in the sensor output. It is typically modeled as the sum of a constant turn-on bias and a slowly time-varying component known as bias instability or in-run bias. This time-varying component is crucial to estimate online and is often modeled as a random walk.[21]

- **Scale Factor Error (s_g, s_a):** This is a multiplicative error representing a deviation in the sensor's sensitivity. For a 3-axis sensor, this can be a full matrix including cross-axis sensitivity terms, which measure how an input on one axis affects the output on another.[21],[23]

- **Noise (n_g, n_a):**This is a high-frequency, zero-mean random fluctuation in the sensor output. It is typically modeled as additive white Gaussian noise, characterized by its noise density. When integrated, this noise leads to an error that grows with the square root of time, known as Angle Random Walk (ARW) for gyroscopes and Velocity Random Walk (VRW) for accelerometers.[21]

Accurate VINS performance relies on the online estimation of these error terms, particularly the biases, which are included as part of the state vector in the filter or optimizer.[24]

IMU Pre-integration A critical innovation for modern tightly-coupled VINS is IMU pre-integration. IMUs typically provide measurements at a much higher frequency (e.g., 200–1000 Hz) than cameras (e.g., 20–30 Hz).[20] Including a state for every IMU measurement in an optimization-based framework would be computationally prohibitive. IMU pre-integration elegantly solves this problem by analytically combining all the inertial measurements between two consecutive camera keyframes into a single relative motion constraint. This process integrates the IMU measurements in the local body frame. A crucial aspect of this formulation is that these pre-integrated terms are expressed relative to the state at the first keyframe and are only dependent on the IMU biases, not the absolute orientation, velocity, or position. This means that if the optimization algorithm updates its estimate of the IMU biases, the pre-integration constraint can be efficiently re-evaluated using first-order corrections without needing to re-propagate all the raw IMU measurements from scratch. This makes tightly-coupled, optimization-based VINS computationally feasible.

4.2.2 The Visual Sensor: Monocular vs. Stereo Configurations

The choice of camera configuration is a fundamental design decision in VINS, with significant trade-offs between cost, complexity, and performance.[25]

Monocular VINS A monocular VINS uses a single camera. The primary advantages of this configuration are its minimal hardware cost, low weight and power consumption, and simplicity, making it an ideal choice for small, payload-constrained UAVs.[26] The fundamental limitation of a monocular camera is its inherent inability to observe absolute scale from a single 2D image. It can only determine the 3D structure of the environment and its own trajectory up to an unknown scale factor.[27] This is a geometric reality: a single projection provides information about the bearing of a point in space, but not its distance.[26] The fusion with an IMU is what elevates a monocular system from a relative, scale-less estimator to a full, metric state estimator. The IMU's accelerometer provides measurements of specific force (including gravity) in absolute metric units (m/s^2). By tightly coupling these metric inertial measurements with the scaled visual geometry from the camera, the VINS estimator can observe and resolve the absolute scale of the trajectory and the map.[26] This observability, however, is not instantaneous. It requires the UAV to undergo sufficient acceleration during an initialization phase for the scale to become well-constrained.[26]

Stereo VINS A stereo VINS employs a pair of cameras with a fixed, known baseline. The primary advantage of this setup is its ability to perceive depth directly and instantaneously. By identifying the same feature in both the left and right images, its 3D position can be calculated through triangulation.[25] This provides an absolute scale for the visual measurements from the very first frame, without requiring any specific motion for initialization.[27] This leads to generally more robust and accurate state estimation.[25] The main disadvantages are the increased hardware cost, size, weight, and complexity.[25] A stereo rig requires precise extrinsic calibration to know the exact transformation between the two cameras, and the system must bear the computational load of processing two image streams.[25] Furthermore, the effectiveness of stereo vision is range-dependent. As the distance to observed features becomes very large relative to the camera baseline, the stereo system's depth estimation capability degrades, and it effectively degenerates to the monocular case.[27]

4.3 Core VINS Architectures: Fusion and Estimation Strategies

The design of a VINS algorithm is defined by two fundamental architectural choices: the method used to fuse data from the visual and inertial sensors, and the mathematical framework used to estimate the state. These choices determine the system's accuracy, robustness, and computational profile.

4.3.1 Data Fusion Paradigms: Loosely-Coupled vs. Tightly-Coupled Approaches

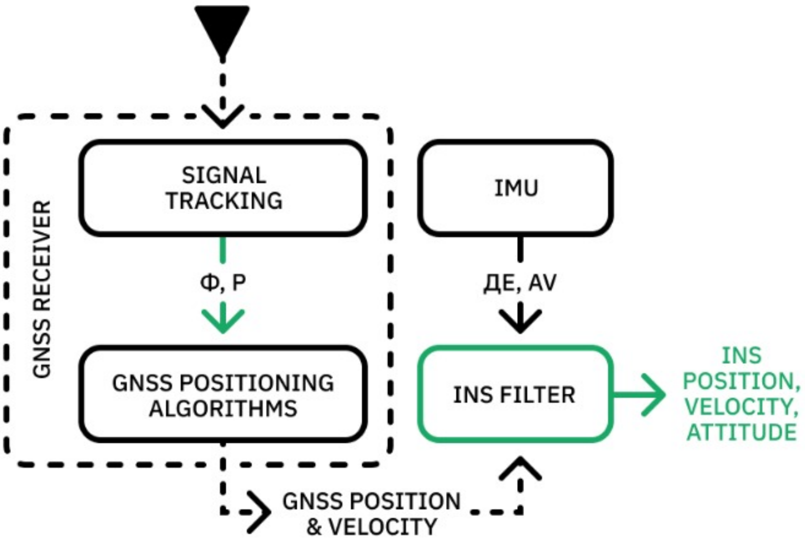


Figure 4: Loosely-coupled architecture which uses an IMU and a GNSS receiver. Similarly, instead of a GNSS receiver, a VINS system can use a visual odometry system.[28]

Loosely-Coupled VINS In a loosely-coupled architecture, the visual and inertial components function as independent estimation modules that are fused at a high level.[28]

The typical data flow involves a visual odometry or VSLAM system producing a 6-DOF pose estimate, while the IMU measurements are integrated to produce a separate pose estimate. A final fusion filter then combines these two independent pose solutions to generate the final state estimate.[28] The primary advantage of this approach is its modularity and simplicity.[28] However, this simplicity comes at the cost of performance. Loosely-coupled fusion is inherently sub-optimal because it discards the rich statistical correlations that exist between the raw visual features and the raw inertial measurements.[29] If the vision system fails temporarily, it provides no output to the fusion filter, forcing the system to rely solely on the drifting IMU integration.[28]

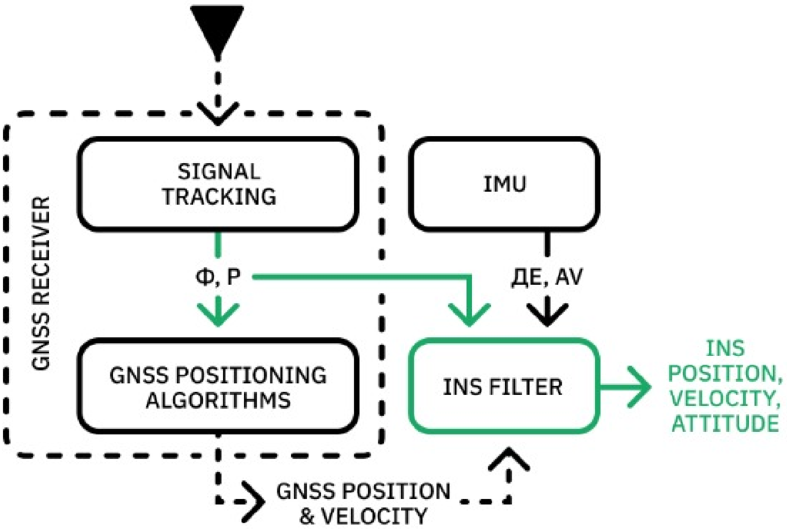


Figure 5: Tightly-coupled architecture which uses an IMU and a GNSS receiver. The GNSS system can be replaced with a visual odometry system.[28]

Tightly-Coupled VINS In contrast, a tightly-coupled architecture is the dominant paradigm in modern, high-performance VINS.[29] This approach establishes a single, unified estimation framework that directly fuses the raw measurements from both sensors.[28] The state estimator, whether a filter or an optimizer, processes visual feature measurements and raw IMU measurements simultaneously within a joint probabilistic model.[30] The principal benefit of tight coupling is superior accuracy and robustness. By considering all raw measurements in a single estimation problem, the framework can fully exploit the statistical correlations between them, leading to a more constrained and optimal solution.[29] The ability to leverage partial information is a key differentiator; even if only a few features are tracked, a tightly-coupled system can use that limited information to mitigate drift, whereas a loosely-coupled system might declare a total tracking failure.[28] The main drawback of this approach is its significantly higher implementation complexity.[28]

4.3.2 Back-End State Estimation: Filtering vs. Optimization

The "back-end" of a VINS is the core computational engine that performs the state estimation. The two primary approaches are filtering and optimization (also known as smoothing).[31]

Filtering-Based VINS Filtering-based methods recursively estimate the current state of the system by incorporating measurements as they arrive. The most prevalent method in this category is the Extended Kalman Filter (EKF).[32] The EKF operates in a continuous two-step cycle:

1. **Prediction (or Propagation):** The system's dynamic model, driven by the IMU measurements, is used to propagate the state estimate and its associated uncertainty forward in time.[32]
2. **Update (or Correction):** When a new measurement from the camera becomes available, the filter compares the actual measurement with a predicted measurement. The difference, or "innovation," is used to correct the state estimate and reduce its uncertainty.[32] .

The primary advantages of filtering-based VINS are their computational efficiency and low memory footprint, making them well-suited for resource-constrained UAVs.[31] However, their main drawback lies in the linearization required by the EKF, which can introduce significant errors and lead to filter inconsistency, where the filter becomes erroneously overconfident in an inaccurate state estimate.[33]

Optimization-Based VINS Optimization-based methods, also known as smoothing methods, have become the standard for high-accuracy VINS. They formulate the state estimation problem as a large-scale nonlinear least-squares minimization. The objective is to find the trajectory and map structure that best explain all available sensor measurements over a period of time by minimizing a cost function. This joint optimization is a generalization of the Bundle Adjustment (BA) technique from computer vision.[34]

The cost function is typically a sum of weighted squared errors from visual and inertial sources.[30] Since optimizing over the entire trajectory is computationally infeasible, these methods employ a sliding window approach, restricting the optimization to a bounded window of the most recent states and measurements.[30] Old states are marginalized, summarizing their information as a probabilistic prior on the remaining states to maintain fixed computational complexity.[30] The principal advantage of optimization-based methods is their superior accuracy. By considering a batch of measurements simultaneously, they can iteratively re-linearize the problem, which significantly mitigates the single-point linearization errors that plague the EKF.[31]

4.4 A Review of Prominent VINS Algorithms

The theoretical concepts of coupling and estimation have been embodied in several influential, open-source VINS algorithms that have defined the state-of-the-art. This section provides a detailed review of four such systems.

4.4.1 MSCKF: The Multi-State Constraint Kalman Filter

The Multi-State Constraint Kalman Filter (MSCKF) is a seminal work in the VINS literature, representing a highly efficient, tightly-coupled, filter-based approach. It is built upon the Extended Kalman Filter (EKF) framework for recursive state estimation[35].

The key innovation of MSCKF lies in its novel measurement update strategy, which cleverly avoids the computational burden of adding 3D feature positions to the filter's state vector. Instead, it maintains a sliding window of recent camera poses within the state vector. When a feature is tracked across several of these poses, a geometric constraint is formed that relates them. This multi-pose constraint is then used to perform a single, efficient EKF update[36].

The primary strength of this ”structureless” approach is its computational efficiency, with a complexity that is linear in the number of tracked features. This makes MSCKF well-suited for computationally-constrained platforms. However, its accuracy is generally lower than optimization-based methods, and its performance can be sensitive to tuning parameters[37]. Furthermore, its delayed update mechanism—where a feature's constraint is only processed after it is no longer tracked—can be suboptimal for long-duration missions[38].

4.4.2 OKVIS: Open Keyframe-based Visual-Inertial SLAM

OKVIS (Open Keyframe-based Visual-Inertial SLAM) is a landmark system that demonstrates the power of a tightly-coupled, optimization-based approach for visual-inertial odometry. It is designed primarily for stereo camera setups but also supports monocular configurations. The core of OKVIS is a nonlinear optimization framework that operates on a bounded-size sliding window of selected keyframes[30]. Its key innovation is the formulation of a single, joint probabilistic cost function that rigorously combines errors from both sensor modalities, including a visual reprojection error term and an inertial error term[30]. To ensure real-time performance, old states are marginalized out, converting their informational contribution into a probabilistic prior on the remaining states[30]. OKVIS was one of the first widely available open-source systems to showcase the superior accuracy achievable with a tightly-coupled, optimization-based VINS[39]. However, its performance is critically dependent on high-quality sensor calibration and precise hardware-level time synchronization between the camera and IMU.

4.4.3 VINS-Mono: A Robust and Versatile Monocular VINS

VINS-Mono is a complete and versatile monocular visual-inertial state estimation system that has become one of the most popular and widely used frameworks in the field. It is a tightly-coupled, optimization-based system renowned for its robustness and comprehensive feature set, which includes automatic initialization, online extrinsic calibration, and loop closure for global drift correction.

A standout feature of VINS-Mono is its robust initialization procedure that can bootstrap the system from unknown moving states[26]. The core of the system is a sliding-window nonlinear optimization for visual-inertial odometry. Beyond local odometry, VINS-Mono incorporates a loop detection module and performs a 4-DOF pose graph optimization to correct for accumulated drift over the entire trajectory, ensuring global consistency[26].

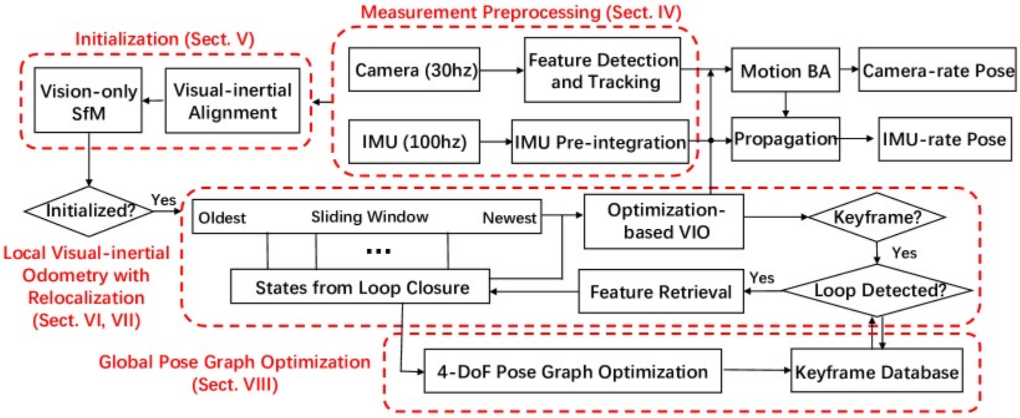


Figure 6: A block diagram illustrating the full pipeline of the visual-inertial state estimator.[26]

The major contribution of VINS-Mono was providing a complete, robust, and easy-to-use open-source package that performs exceptionally well across a wide range of applications, from aerial robots to standard smartphones.[26] Its excellent performance and versatility have established it as a de facto benchmark for the evaluation of new VINS algorithms.[40]

4.4.4 ORB-SLAM3: A Multi-Map Visual-Inertial SLAM System

ORB-SLAM3 represents the current state-of-the-art in full SLAM systems, supporting visual-only and visual-inertial SLAM with monocular, stereo, and RGB-D cameras. ORB-SLAM3 introduces two main novelties. The first is a tightly-integrated visual-inertial SLAM system based on Maximum-a-Posteriori (MAP) estimation, which is applied to all stages of the algorithm, including a novel and highly accurate IMU initialization procedure. This results in a system that is reported to be 2 to 10 times more accurate than previous approaches.

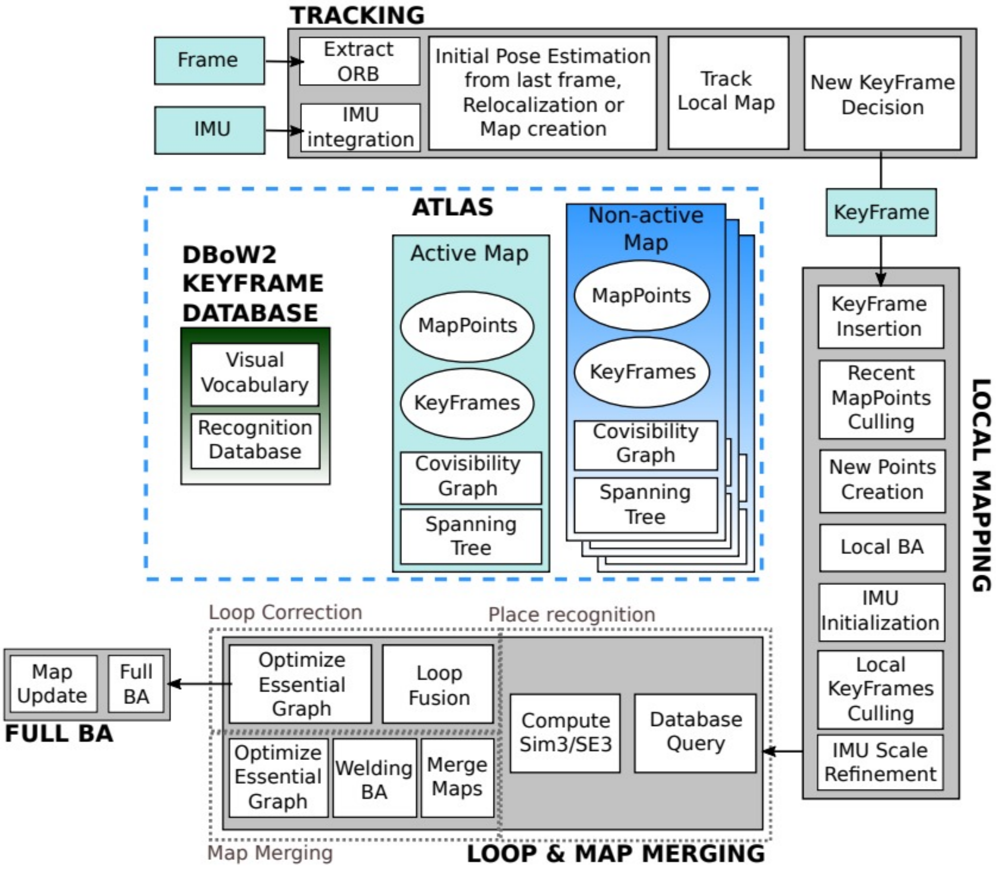


Figure 7: Main system components of ORB-SLAM3.[41]

The second and most significant innovation is its multi-map system, called Atlas.[41] In traditional SLAM systems, a loss of visual tracking is often a catastrophic failure. ORB-SLAM3, however, is able to survive these events. When tracking is lost, it seamlessly initiates a new, independent map. It then continues to operate in this new map until it re-enters a previously mapped area, at which point it robustly merges the new map with the old one, creating a single, globally consistent map. This capability allows ORB-SLAM3 to perform robustly during long-term operation and to reuse and stitch together maps from different sessions.[41]

Table 1: Comparison of Prominent VINS Algorithms

Algorithm	**Core Method**	**Coupling**	**Camera Support**	**Key Innovations & Strengths**	**Key Limitations & Challenges**
MSCKF	Filter-based (EKF)	Tightly	Monocular, Stereo	Efficiency: Avoids adding features to state vector, linear complexity. Consistency: Good consistency properties in later versions.	Accuracy: Generally less accurate than optimization methods. Sensitive to tuning. Delayed update can be suboptimal.
OKVIS	Optimization-based (BA)	Tightly	Stereo, Monocular	Accuracy: High accuracy via joint nonlinear optimization of visual and inertial errors. Keyframe Paradigm: Bounded complexity for real-time use.	Complexity: Higher computational cost than filters. Highly sensitive to sensor calibration and synchronization.
VINS-Mono	Optimization-based (BA)	Tightly	Monocular	Robustness & Versatility: Excellent all-around performance. Robust Initialization: Can bootstrap from unknown states. Includes loop closure and online calibration.	Monocular setup requires motion for initialization. Performance can degrade in challenging lighting without extensions (e.g., PC-VINS-Mono).
ORB-SLAM3	Optimization-based (MAP/BA)	Tightly	Monocular, Stereo, RGB-D	Long-Term Autonomy: Multi-map system (*Atlas*) survives tracking loss and enables map reuse. Accuracy: MAP-based estimation provides state-of-the-art accuracy.	Complexity: Highly complex codebase. Primarily a SLAM system, may be overkill for pure odometry tasks.

4.5 Challenges and Future Research Directions

While VINS has matured into a powerful technology, its application to autonomous UAVs presents a unique set of challenges. Concurrently, the field is rapidly evolving, with several key research trends pointing toward the next generation of navigation systems.

4.5.1 Platform-Specific Challenges for UAVs

- **High-Speed Motion and Aggressive Maneuvers:** UAVs are capable of executing highly dynamic trajectories involving large accelerations and rapid rotations. Such motion poses a severe challenge to the visual component of VINS, inducing large apparent motion in the image plane and significant motion blur, which can cause traditional feature trackers to fail.[32]

- **Computational and Power Constraints:** UAVs are fundamentally payload-constrained systems. VINS algorithms must therefore be computationally efficient to run on lightweight, low-power embedded hardware. This creates a persistent tension between algorithm accuracy and computational cost, forcing developers to make critical trade-offs between estimation performance and the operational constraints of the platform.[31]

- **Sensor Robustness in Challenging Environments:** The performance of a VINS is inextricably linked to the quality of its sensor data. UAVs often operate in visually challenging environments where camera performance can be severely degraded. Conditions such as poor or dynamically changing illumination, visually textureless surfaces, and adverse weather can cause feature detection and tracking algorithms to fail.[42]

4.5.2 Emerging Trends and Open Problems

The solutions being developed to address these challenges indicate a paradigm shift in the VINS field. Progress is no longer solely about refining existing geometric estimators but about integrating new technologies and concepts to build more comprehensive and intelligent perceptual systems.

- **Integration of Deep Learning:**Deep learning is poised to revolutionize many aspects of the VINS pipeline. This ranges from replacing individual classical components with learned alternatives—for instance, using deep neural networks to detect more robust features [43]—to developing end-to-end systems that directly regress pose from sensor inputs. A more profound trend is the use of deep learning for semantic scene understanding, enabling a UAV to distinguish between static background (useful for localization) and dynamic objects (which violate the static world assumption).[31]

- **Achieving Long-Term Robustness and Life-Long Mapping:** The ambition of the field is moving beyond short-term odometry towards creating systems capable of long-term, persistent autonomy. This requires solving several hard problems, such as robustly handling tracking failures, as demonstrated by the multi-map system in ORB-SLAM3, and adapting to changes in the environment over time.[31]
- **Multi-Sensor and Multi-Modal Fusion:** While the fusion of a camera and an IMU is powerful, robustness can be further enhanced by incorporating additional sensor modalities. Fusing VINS with LiDAR can provide dense, accurate geometric information that is immune to lighting conditions.[44] Integrating radio-based ranging measurements, such as 5G Time-of-Arrival (ToA), can provide absolute position updates to correct for drift.[45] Perhaps most promising for agile UAVs is the fusion with event cameras. These bio-inspired sensors asynchronously report per-pixel brightness changes, giving them an extremely high temporal resolution, high dynamic range, and very low latency, making them exceptionally well-suited for robust perception during the high-speed, aggressive maneuvers where standard cameras fail.[46]

This evolution indicates that the future of VINS lies in a holistic approach, integrating advanced techniques from machine learning, novel sensor engineering, and large-scale data management to create perceptual systems that are not only accurate but also intelligent, adaptable, and truly robust in the complexities of the real world.

5 Methodology

5.1 Phase 1: Problem Formulation and Requirement Analysis

This initial stage lays the foundation for the entire project by ensuring that the problem is well-defined, relevant, and achievable. It establishes a clear direction and avoids scope creep later.

- **Literature Review and Background Research:** Conduct a systematic review of academic papers, technical reports, and existing implementations of Visual-Inertial Navigation Systems (VINS), drone navigation frameworks, and swarm robotics.

Purpose: Understand current capabilities, identify common limitations (e.g., drift errors, processing latency, environmental constraints), and recognize trends in drone-based navigation research.

Outcome: A consolidated knowledge base to guide technical decisions.

- **Gap Identification and Problem Statement:** Based on literature findings, identify gaps in current research, such as limited performance in GPS-denied environments or scalability issues in swarm navigation. Formulate a clear problem statement that encapsulates these gaps.

- **Requirement Gathering:** Define functional requirements (e.g., real-time localization, marker-based pose estimation, swarm coordination) and non-functional requirements (e.g., low computational latency, high robustness against sensor noise).

- **Project Proposal Development:** Prepare a formal proposal containing objectives, methodology, scope, anticipated challenges, resource needs, and success metrics. This document serves as the agreement point before moving into technical work.

5.2 Phase 2: Resource Gathering

This phase ensures that all essential hardware, datasets, algorithms, and software platforms are prepared before development begins. It prevents delays due to missing components later.

- **Model Selection:** Review and shortlist open-source and proprietary VINS algorithms such as VINS-Mono, ORB-SLAM3, or VINS Fusion.

  Selection Criteria: Accuracy, computational efficiency, ease of integration, hardware requirements.

- **Dataset Acquisition:** Obtain benchmark datasets (e.g., EuRoC MAV, KITTI) for algorithm testing and training. Additionally, plan custom dataset collection using drone-mounted cameras and IMUs in varied environments.

  Purpose: Ensure the models are tested in scenarios similar to real deployment conditions.

- **Hardware Resource Gathering:** Identify and procure all necessary hardware components such as drones, cameras, IMUs, microcontrollers, onboard computers (e.g., Jetson Nano, Raspberry Pi), servos, and communication modules. Ensure all devices meet the project's technical specifications and are compatible with planned software platforms.

Purpose: Prevent hardware shortages or mismatches during integration and testing.

- **Tool and Platform Setup:** Install and configure simulation platforms (Gazebo, AirSim, ROS), development environments (Python, C++), and supporting libraries (OpenCV, Eigen, g2o).

  Hardware setup: Ensure the drone's onboard computer and peripheral sensors are functional and calibrated.

5.3 Phase 3: Model Evaluation

Here, shortlisted models are rigorously tested under controlled conditions to determine the best candidate for real-world integration.

- **Platform-Based Testing:** Deploy each selected VINS algorithm in simulated environments, replicating varied real-world conditions such as poor lighting, dynamic objects, and fast motion.
- **Performance Analysis:** Evaluate metrics such as Absolute Trajectory Error (ATE), Relative Pose Error (RPE), frame processing rate (FPS), and robustness under sensor noise.

  Purpose: Identify which model consistently meets or exceeds performance thresholds.
- **Comparative Evaluation:** Rank models according to performance scores and choose the most suitable candidate, justifying the selection with quantitative data.

5.4 Phase 4: Drone Integration

This stage focuses on implementing the chosen VINS system into a physical drone and testing its operational performance.

- **Drone Assembly and Implementation:** Assemble the drone hardware (frames, motors, sensors, communication modules) and integrate all electronic components. Ensure proper wiring, power distribution, and mechanical stability before software integration.
- **Algorithm Implementation:** Port the selected VINS model into the drone's onboard computing system (e.g., NVIDIA Jetson Nano, Raspberry Pi 4) while ensuring real-time execution.

- **VINS and Flight Controller Integration:** Interface the VINS output with the drone's flight control system (e.g., PX4, ArduPilot) so that navigation decisions are directly informed by visual-inertial data.
- **Flight Testing:** Conduct a series of indoor and outdoor test flights in different conditions:
  - Low light
  - High wind
  - GPS-denied areas

  Record performance metrics and operational stability for further refinement.

5.5 Phase 5: Model Improvements and Drone Swarm Exploration

Once a working prototype is validated, efforts shift toward improving robustness and extending functionality to swarm scenarios.

- **Limitation Analysis:** Use flight logs and recorded datasets to identify recurring problems such as drift, latency, or unstable tracking.
- **System Optimization:** Apply algorithmic improvements (e.g., better feature tracking, adaptive IMU calibration) and hardware enhancements (e.g., more powerful onboard processors).
- **Pose Estimation with Markers:** Integrate marker-based localization (e.g., AprilTags, ArUco markers) to assist pose estimation when visual features are scarce.
- **Swarm Coordination Exploration:** Begin experimental swarm operations, focusing on:
  - Inter-drone communication
  - Formation control
  - Collaborative mapping and navigation

5.6 Phase 6: Documentation and Dissemination

Final phase ensures that all technical, operational, and research details are preserved and communicated effectively.

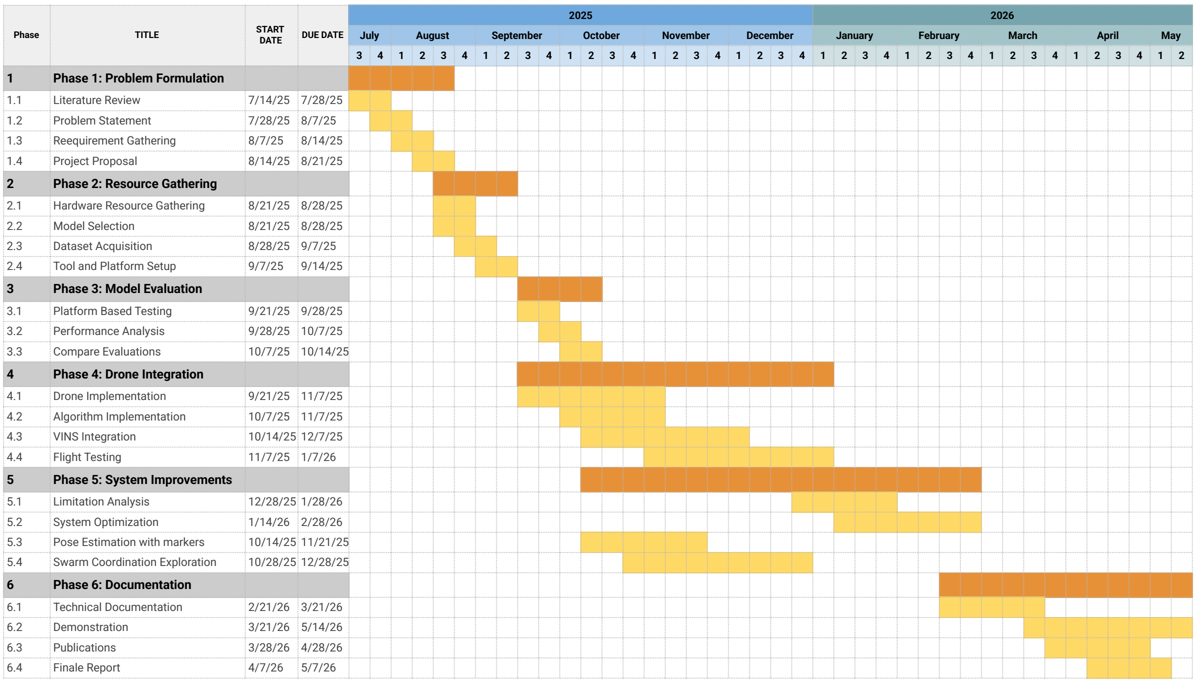
- **Technical Documentation:** Prepare exhaustive documentation covering system architecture, integration steps, source code references, and testing methodologies.
- **Demonstration:** Organize a live or recorded demonstration showing the system's capabilities in real flight and/or swarm scenarios.
- **Research Publications:** Compile results into conference papers or journal articles to share findings with the academic and professional community.
- **Final Reports:** Produce a complete project report including methodology, results, discussion, limitations, and future work recommendations.

5.7 Summary of Phases

Phase	**Main Focus**	**Expected Outcomes**
1. Problem Formulation	Define problem, requirements, proposal	Clear problem statement, project proposal, defined scope
2. Resource Gathering	Collect models, datasets, platforms	Ready-to-use algorithms, datasets, and simulation environments
3. Model Evaluation	Test and compare VINS models	Performance metrics, best model selected
4. Drone Integration	Implement and test VINS on hardware	Functional VINS-enabled drone with tested navigation
5. Model Improvement & Swarm Exploration	Optimize system, extend to swarm	Improved robustness, initial swarm coordination results
6. Documentation & Dissemination	Reporting and sharing findings	Technical documentation, publications, final project report

Table 2: Summary of Methodology Phases with Focus and Expected Outcomes.

6 Research Timeline



Phase	TITLE	START DATE	DUE DATE
1	**Phase 1: Problem Formulation**		
1.1	Literature Review	7/14/25	7/28/25
1.2	Problem Statement	7/28/25	8/7/25
1.3	Reeqirement Gathering	8/7/25	8/14/25
1.4	Project Proposal	8/14/25	8/21/25
2	**Phase 2: Resource Gathering**		
2.1	Hardware Resource Gathering	8/21/25	8/28/25
2.2	Model Selection	8/21/25	8/28/25
2.3	Dataset Acquisition	8/28/25	9/7/25
2.4	Tool and Platform Setup	9/7/25	9/14/25
3	**Phase 3: Model Evaluation**		
3.1	Platform Based Testing	9/21/25	9/28/25
3.2	Performance Analysis	9/28/25	10/7/25
3.3	Compare Evaluations	10/7/25	10/14/25
4	**Phase 4: Drone Integration**		
4.1	Drone Implementation	9/21/25	11/7/25
4.2	Algorithm Implementation	10/7/25	11/7/25
4.3	VINS Integration	10/14/25	12/7/25
4.4	Flight Testing	11/7/25	1/7/26
5	**Phase 5: System Improvements**		
5.1	Limitation Analysis	12/28/25	1/28/26
5.2	System Optimization	1/14/26	2/28/26
5.3	Pose Estimation with markers	10/14/25	11/21/25
5.4	Swarm Coordination Exploration	10/28/25	12/28/25
6	**Phase 6: Documentation**		
6.1	Technical Documentation	2/21/26	3/21/26
6.2	Demonstration	3/21/26	5/14/26
6.3	Publications	3/28/26	4/28/26
6.4	Finale Report	4/7/26	5/7/26

Figure 8: Research Timeline

7 Conclusion

This project proposal has outlined a comprehensive approach to developing a robust Visual-Inertial Navigation System (VINS) for autonomous UAV operations in GPS-restricted environments. The expansion of UAV applications into complex settings like dense urban areas, indoor facilities, and under-canopy forests highlights the critical limitations of GPS, which can suffer from signal blockages, multipath propagation, and intentional interference. Our research demonstrates that VINS, by fusing data from cameras and Inertial Measurement Units (IMUs), provides a compelling solution to this challenge, enabling accurate and reliable state estimation without reliance on external satellite signals.

The project methodology is structured in a series of phases, beginning with a thorough literature review to identify key challenges such as IMU drift, visual tracking failures in degraded scenes, and computational demands. By evaluating prominent VINS algorithms like MSCKF, OKVIS, VINS-Mono, and ORB-SLAM3, we have established a clear path to selecting and implementing a system that balances accuracy, robustness, and computational efficiency. The chosen system will be integrated into a physical UAV platform, and its performance will be rigorously tested in various challenging scenarios to ensure it meets real-world operational requirements.

Ultimately, the successful implementation of this project will provide a critical enabling technology for the next generation of autonomous drones. By addressing the "last mile" navigation problem, our work will unlock the full potential of UAVs for high-impact applications such as precision delivery, detailed inspections, and search-and-rescue missions, moving beyond the constraints of line-of-sight operations. The exploration of swarm coordination in later phases will further extend the system's capabilities, paving the way for collaborative autonomous missions. This research is not merely a technical exercise but a crucial step toward building intelligent, adaptable, and truly robust aerial systems for a wide range of strategic and commercial applications.

8 References

[1] F. Toscano et al., "Unmanned aerial vehicle for precision agriculture: A review," *IEEE Access*, pp. 1–1, Jan. 2024. DOI: 10.1109/access.2024.3401018

[2] *List of unmanned aerial vehicle applications*, Accessed: Apr. 12, 2020, 2020. [Online]. Available: https://en.wikipedia.org/wiki/List_of_unmanned_aerial_vehicle_applications

[3] Y. Cong et al., "Adaptive covariance matrix for uav-based visual-inertial navigation systems using gaussian formulas," *Sensors*, vol. 25, no. 15, p. 4745, Jan. 2025. DOI: 10.3390/s25154745

[4] M. Peretic et al., "Statistical analysis of gnss multipath errors in urban canyons," in *Proc. IEEE/ION Position Location and Navigation Symp. (PLANS)*, Apr. 2025, pp. 1216–1225. DOI: 10.1109/plans61210.2025.11028411

[5] W. Wen, X. Bai, and L.-T. Hsu, "3d vision aided gnss real-time kinematic positioning for autonomous systems in urban canyons," *Navigation*, vol. 70, no. 3, navi.590–navi.590, Jan. 2023. DOI: 10.33012/navi.590

[6] W. Wen, X. Bai, and L.-T. Hsu, *3d vision aided gnss real-time kinematic positioning for autonomous systems in urban canyons*, Accessed: 2025-08-20, 2023. [Online]. Available: https://www.polyu.edu.hk/aae/ipn-lab/us/publications/Fullpaper/3D%20Vision%20Aided%20GNSS%20Real-Time%20Kinematic%20Positioning%20for%20Autonomous%20Systems%20in%20Urban%20Canyons.pdf

[7] L. Zhao, W. Wang, Q. He, L. Yan, and X. Li, "Visual–inertial autonomous uav navigation in complex illumination and highly cluttered under-canopy environments," *Drones*, vol. 9, no. 1, pp. 27–27, Jan. 2025. DOI: 10.3390/drones9010027

[8] H. A. Hashim, "Advances in uav avionics systems architecture, classification and integration: A comprehensive review and future perspectives," *Results in Engineering*, vol. 25, pp. 103 786–103 786, Dec. 2024. DOI: 10.1016/j.rineng.2024.103786

[9] *Moving away from gps to a multi-sensor, inertial-centered architecture*, Accessed: Aug. 18, 2025, Jun. 2025. [Online]. Available: https://www.geoweeknews.com/news/moving-away-from-gps-to-a-multi-sensor-inertial-centered-architecture

[10] J.-C. Lee et al., "Landmark-based scale estimation and correction of visual inertial odometry for vtol uavs in a gps-denied environment," *Sensors*, vol. 22, no. 24, p. 9654, 2022.

[11] Palantir Technologies, *The future of drone navigation*, Accessed: 2025-08-20, 2024. [Online]. Available: https://blog.palantir.com/the-future-of-drone-navigation-7236075fdedf

[12] W. Huang, W. Wan, and H. Liu, "Optimization-based online initialization and calibration of monocular visual-inertial odometry considering spatial-temporal constraints," *Sensors*, vol. 21, no. 8, p. 2673, 2021.

[13] Y. Liu, Z. Feng, H. Zhang, and W. Dong, "Post-integration based point-line feature visual slam in low-texture environments," *Scientific Reports*, vol. 15, p. 14 606, 2025.

[14] P. Liu, X. Zuo, V. Larsson, and M. Pollefeys, "Mba-vo: Motion blur aware visual odometry," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5550–5559.

[15] Y. Gao et al., "A simultaneous control, localization, and mapping system for uavs," *Drones*, vol. 9, no. 1, p. 69, 2025, Accessed: Aug. 14, 2025. DOI: 10.3390/drones9010069

[16] *Vision-based localization methods under gps-denied conditions*, Accessed: Aug. 14, 2025, 2022. [Online]. Available: https://arxiv.org/pdf/2211.11988

[17] *Viavi launches vins to tackle rising gps jamming for uav navigation*, Accessed: Aug. 14, 2025, 2025. [Online]. Available: https://www.nasdaq.com/articles/viavi-launches-vins-tackle-rising-gps-jamming-uav-navigation

[18] T. Qin, P. Li, and S. Shen, *Vins-mono: A robust and versatile monocular visual-inertial state estimator*, Accessed: Aug. 14, 2025, 2017. [Online]. Available: https://arxiv.org/abs/1708.03852

[19] *Visual-inertial navigation system*, Accessed: Aug. 14, 2025, 2025. [Online]. Available: https://www.uavnavigation.com/company/blog/visual-inertial-navigation-system

[20] *A comprehensive introduction of visual-inertial navigation*, Accessed: Aug. 14, 2025, 2023. [Online]. Available: `https://arxiv.org/pdf/2307.11758`

[21] *Inertial measurement unit (imu) — an introduction*, Accessed: Aug. 14, 2025, 2025. [Online]. Available: `https://www.advancednavigation.com/tech-articles/inertial-measurement-unit-imu-an-introduction/`

[22] *Improving monocular visual-inertial initialization with structureless visual-inertial bundle adjustment*, Accessed: Aug. 14, 2025, 2025. [Online]. Available: `https://arxiv.org/html/2502.16598v1`

[23] Y. Wu et al., "Mems imu error mitigation using rotation modulation technique," *Sensors*, vol. 16, no. 12, p. 2017, 2016, Accessed: Aug. 14, 2025. DOI: `10.3390/s16122017`

[24] T. Rehbinder et al., "An evaluation of mems-imu performance on the absolute trajectory error of visual-inertial navigation system," *Micromachines*, vol. 13, no. 4, p. 602, 2022, Accessed: Aug. 14, 2025. DOI: `10.3390/mi13040602`

[25] A. Jain, *Monocular vs stereo vs monochrome camera*, Accessed: Aug. 14, 2025, 2023. [Online]. Available: `https://medium.com/@abhishekjainindore24/monocular-vs-stereo-vs-monochrome-camera-16612bf4358b`

[26] T. Qin, P. Li, and S. Shen, *Vins-mono: A robust and versatile monocular visual-inertial state estimator*, Accessed: Aug. 14, 2025, 2017. [Online]. Available: `https://ar5iv.labs.arxiv.org/html/1708.03852`

[27] A. Singh, *Visual odometry full tutorial*, Accessed: Aug. 14, 2025, 2023. [Online]. Available: `https://avisingh599.github.io/vision/visual-odometry-full/`

[28] *Loosely coupled & tightly coupled ins & gnss [2024 guide]*, Accessed: Aug. 14, 2025, 2024. [Online]. Available: `https://pointonenav.com/news/loose-vs-tight-coupling-gnss/`

[29] G. Huang et al., *Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry*, Accessed: Aug. 14, 2025, 2021. [Online]. Available: `https://udel.edu/~ghuang/icra21-vins-workshop/papers/06-Cioffi_global-VIO.pdf`

[30] *Keyframe-based visual-inertial odometry using nonlinear optimization*, Accessed: Aug. 14, 2025, 2014. [Online]. Available: `https://www.researchgate.net/publication/265683241_Keyframe-Based_Visual-Inertial_Odometry_Using_Nonlinear_Optimization`

[31] X. Hu et al., “A review of visual-inertial simultaneous localization and mapping,” *Robotics*, vol. 7, no. 3, p. 45, 2018, Accessed: Aug. 14, 2025. DOI: 10.3390/robotics7030045

[32] *Extended kalman filter navigation overview and tuning*, Accessed: Aug. 14, 2025, 2025. [Online]. Available: https://ardupilot.org/dev/docs/extended-kalman-filter.html

[33] *Sp-vio: Robust and efficient filter-based visual inertial odometry with state transformation model and pose-only visual description*, Accessed: Aug. 14, 2025, 2024. [Online]. Available: https://arxiv.org/html/2411.07551v1

[34] B. Triggs et al., *Bundle adjustment — a modern synthesis*, Accessed: Aug. 14, 2025, 2000. [Online]. Available: https://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/Triggs00.pdf

[35] Y. E. Kalmanis et al., *A multi-state constraint kalman filter for vision-aided inertial navigation (presentation)*, Accessed: Aug. 14, 2025, 2023. [Online]. Available: https://cse.sc.edu/~yiannisr/774/2023/MSCKF%20Presentation.pptx

[36] A. Mourikis and S. Roumeliotis, *A multi-state constraint kalman filter for vision-aided inertial navigation*, Accessed: Aug. 14, 2025, 2007. [Online]. Available: https://www.researchgate.net/publication/224705670_A_Multi-State_Constraint_Kalman_Filter_for_Vision-Aided_Inertial_Navigation

[37] *The battle for filter supremacy: A comparative study of the multi-state constraint kalman filter and the sliding window filter*, Accessed: Aug. 14, 2025, 2015. [Online]. Available: https://www.researchgate.net/publication/279752987_The_Battle_for_Filter_Supremacy_A_Comparative_Study_of_the_Multi-State_Constraint_Kalman_Filter_and_the_Sliding_Window_Filter

[38] *An immediate update strategy of multi-state constraint kalman filter*, Accessed: Aug. 14, 2025, 2025. [Online]. Available: https://www.themoonlight.io/en/review/an-immediate-update-strategy-of-multi-state-constraint-kalman-filter

[39] *Accurate monocular visual-inertial slam using a map-assisted ekf approach*, Accessed: Aug. 14, 2025, 2017. [Online]. Available: https://arxiv.org/pdf/1706.03648

[40] T. Qin, P. Li, and S. Shen, *Vins-mono: A robust and versatile monocular visual-inertial state estimator*, Accessed: Aug. 14, 2025, 2017. [Online]. Available: https://www.semanticscholar.org/paper/VINS-Mono%3A-A-Robust-and-Versatile-Monocular-State-Qin-Li/9da965529ee3da77178ed99cf13d97be0fa85f6e

[41] *Visual-inertial odometry survey (arxiv:2007.11898v2)*, Accessed: Aug. 14, 2025, 2021. [Online]. Available: https://arxiv.org/abs/2007.11898

[42] L. Zhao et al., "Visual–inertial autonomous uav navigation in complex illumination and highly cluttered under-canopy environments," *Drones*, vol. 9, no. 1, p. 27, 2025, Accessed: Aug. 14, 2025. DOI: 10.3390/drones9010027

[43] *Augmenting orb-slam3 with deep features, adaptive nms, and learning-based loop closure*, Accessed: Aug. 14, 2025, 2025. [Online]. Available: https://arxiv.org/html/2506.13089v1

[44] *Comparison of loosely coupled and tightly coupled ins/lidar systems*, Accessed: Aug. 14, 2025, 2016. [Online]. Available: https://www.researchgate.net/figure/Comparison-of-loosely-coupled-and-tightly-coupled-INS-LiDAR-systems_tbl2_281612795

[45] *Global slam in visual-inertial systems with 5g time-of-arrival integration*, Accessed: Aug. 14, 2025, 2024. [Online]. Available: https://arxiv.org/html/2412.12406v1

[46] *Visual-inertial odometry of aerial robots*, Accessed: Aug. 14, 2025, 2019. [Online]. Available: https://www.researchgate.net/publication/333678568_Visual-Inertial_Odometry_of_Aerial_Robots