UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Synthesis and Evaluation of an Image-Based Redundancy System for UAV Navigation

Sameer Shaboodien
25002783

Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Dr R. P. Theart

October 2024

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / *25002783*	Handtekening / *SamS*
Voorletters en van / *S Shaboodien*	Datum / *27/10/24*

Abstract

Unmanned Aerial Vehicles (UAVs) rely on the Global Positioning System (GPS) for precise navigation, but vulnerabilities such as jamming and spoofing can jeopardize mission success. This project developed an image-based GPS localization system to enable reliable UAV navigation in GPS-denied environments. The system optimizes the localization pipeline by integrating advanced feature extraction, matching, and planar transformation techniques, achieving radial localization errors below 10% of the UAV's displacement from a reference image. Designed for real-time operation, the system responds within two seconds after GPS signal loss, allowing pilots to maintain effective control. Comprehensive testing across diverse environments demonstrated the system's high accuracy, robustness, and generalizability without requiring environment-specific tuning. The system excelled in low-light and low-overlap conditions, ensuring dependable navigation data delivery in various scenarios. This image-based localization provides a practical and reliable alternative to GPS, enhancing UAV operational safety and effectiveness in both military and civilian applications. With further enhancements, the system is poised for real-world deployment, promising safer and more reliable UAV missions.

Contents

List of Figures

List of Tables

Nomenclature

Variables and Functions

$RMSE$	Root Mean Square Error, representing the average magnitude of errors. In this study, the usage of this metric specifically, and frequently, uses this term to refer to the radial error in GPS estimate, in metres, averaged across the dataset.
d_{radial}	Radial error distance, measuring the distance between estimated and true positions in meters.
x, y	Coordinates of a point in an image or on the ground plane.
θ	Rotation angle between consecutive images in degrees or radians, used for orientation alignment.
T	Transformation matrix representing the planar transformation between images.
H	Homography matrix, specifically mapping points from one image plane to another.
GPS	Global Positioning System, providing geolocation and time information for navigation.
$GNSS$	Global Navigation Satellite System, a generic term for satellite-based navigation systems.
MAE	Mean Absolute Error, measuring the average localization error without directionality.
MI	Mutual Information, measuring the similarity or overlap between images, often used in image matching.

Key Terms and Concepts

Features	Unique keypoints in an image along with their descriptors, used for identifying and uniquely matching points across different images for localization.
Affine Transformation	A planar transformation that preserves points, straight lines, and planes, including translation, scaling, rotation, and shearing.
Descriptors	Numerical values describing the characteristics of keypoints in an image, allowing for effective matching between different images.
Feature Detectors (or Extractors)	Processes for identifying distinct features in an image, which are invariant to changes in scale, rotation, and illumination.
Global Matching	Process of finding similarities and determining pose between entire images, considering full-image context rather than isolated keypoints.
GPS (Global Positioning System)	A satellite navigation system providing geolocation and time information. Vulnerable to jamming and spoofing, posing reliability issues for UAV navigation.
Homography	A planar transformation mapping points from one image plane to another in 2D space. Used in image processing to describe transformations like rotation, translation, scaling, and shearing.
Homography Estimation	The process of calculating the homography matrix that defines the transformation between two images for alignment and reliable localization.
Keypoints	Specific points in an image used to identify features. Typically areas of strong contrast or distinct patterns, enabling reliable matching across different images.
Local Matching	Matching keypoints between images by comparing descriptors, focusing on individual feature descriptors to establish precise localization.
Mutual Information	A metric quantifying the information shared between two images, aiding in assessing the quality of feature matching and overlap similarity.
Planar Transforms	General transformations applied to a plane in 2D space, such as affine transformations, homography, scaling, and shearing, which are crucial for image alignment.
Scaling	A transformation that changes the size of an image or its features without altering its shape, normalizing feature sizes across different

Acronyms and Abbreviations

GPS	Global Positioning System
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MI	Mutual Information
UAV	Unmanned Aerial Vehicle

GNSS Lat-Lon

Chapter 1

Introduction

1.1. Background

Unmanned Aerial Vehicles (UAVs) have become indispensable tools in various sectors, including military operations, surveillance, reconnaissance, and intelligence gathering. In South Africa, UAVs play a crucial role in border monitoring and supporting military missions by providing persistent aerial observation [3]. Their capability to operate in hazardous or inaccessible areas enhances operational effectiveness and safety.

Despite their widespread use, UAVs predominantly rely on Global Navigation Satellite Systems (GNSS) such as the Global Positioning System (GPS) for navigation and positioning. GNSS operates by utilizing a constellation of satellites that transmit precise time-stamped signals to Earth-based receivers. Each receiver calculates its position by measuring the time delay of signals from multiple satellites, requiring data from at least four satellites to determine its three-dimensional location. While GNSS provides essential Positioning, Navigation, and Timing (PNT) information globally, its reliance on weak, line-of-sight satellite signals introduces significant vulnerabilities [4].

In recent years, the vulnerabilities of GNSS to jamming and spoofing have become increasingly pronounced. With the advent of more accessible jamming and spoofing technology, intentional GNSS interference is no longer a complex undertaking CITE XXX. Instances of GNSS jamming and spoofing are particularly prevalent in conflict zones, where adversaries exploit GNSS weaknesses to disrupt or mislead UAV operations. The issue is far from hypothetical; over 1100 daily incidents of aircraft GNSS spoofing alone have been reported worldwide, underscoring the urgency of addressing these vulnerabilities CITE XXX.

The increasing prevalence of GNSS disruptions has direct implications for national security, as UAVs may become disoriented or lost, jeopardizing missions and assets. Loss of GNSS signals can lead to an inability to locate the UAV, compromising control and potentially resulting in the UAV crashing or being captured. This situation emphasizes the critical need for alternative navigation solutions that can operate independently of external signals like GNSS. These systems must be robust, accurate, and adaptable to a wide range of environments, while also being cost-effective and minimizing detectability.

Various alternative navigation methods have been explored to mitigate reliance on Global Navigation Satellite Systems (GNSS). Quantum sensing navigation employs cold atom inertial sensors to achieve superior precision compared to traditional inertial measurement units (IMUs), but it is often prohibitively expensive and bulky CITE XXX. Odometry-based solutions estimate UAV displacement through wheel rotation or accelerometer readings; however, they suffer from significant drift over extended distances, rendering their location estimations unusable for missions exceeding roughly 200 km, depending on their level of precision [5]. Radio Frequency (RF) communication systems can triangulate UAV positions using ground beacons or cellular towers, yet their effectiveness is limited in remote areas and by the Earth's curvature, restricting their operational range to approximately 300 km CITE XXX. Light Detection and Ranging (LIDAR) systems map the ground by emitting laser pulses and measuring their return times to create detailed 3D environmental maps, but they are resource-intensive and emit detectable signals, which are undesirable for stealth operations prevalent in military contexts [6].

Image-based navigation emerges as a promising approach that does not succumb to the aforementioned issues. By leveraging onboard cameras, UAVs can reference images with known telemetry data, either through an acquired database or by capturing images during flight prior to GNSS signal loss, to estimate their position and heading. Image-based navigation relies on planar transformations, which relate two images of the same planar surface taken from different perspectives. These transformations enable the alignment of images from different viewpoints, facilitating the estimation of the UAV's orientation and position relative to reference telemetry data CITE XXX.

This method provides a financially accessible way to navigate in GNSS-denied environments without emitting detectable signals or suffering from drift. It benefits from advancements in computer vision and image processing techniques, which have significantly improved the robustness and accuracy of image processing tasks CITE XXX. SUMMARY OF LIT STUDIES - NOT ENOUGH SOLUTIONS. However, it remains unclear whether image-based systems can effectively operate in the context of UAV navigation. Specifically, it is uncertain if they can generalize to various terrains and operational conditions, such as varying light levels, while also achieving real-time operation and high accuracy. This study aims to create a working pipeline for this context and test it on real-world data, thereby addressing these uncertainties and evaluating the viability of image-based navigation as a redundancy measure to GNSS. XXX - waiting for lit

sources: [6]. LIDAR [7]. RF [5] Odometry [8]. Quantum sensing

xxx The latter method is employed in this study and allows return navigation to base along the outbound path. - MUST BE SOMEWHERE scope should include the IMUs for path finding

1.2. Problem Statement

The increasing frequency of GNSS disruptions due to jamming and spoofing poses a significant threat to UAV operations CITE XXX. Current alternatives to GNSS are either too expensive, bulky, decrease stealth capabilities, or suffer from significant drift over extended periods [8] REFERENCE THE REST. Other solutions in the field of image-based navigation do not XYZ LIT REVIEW. Developing such a solution is essential for enhancing UAV operational resilience in contested or denied environments. XXX - waiting for lit

1.3. Aims and Objectives

1.3.1. Aim

The primary aim of this project is to develop and evaluate an image-based navigation system for UAVs that can accurately estimate position and heading in GNSS-denied environments by leveraging images and telemetry data captured prior to GNSS-denial. This approach enables the UAV to navigate back to base post-GNSS denial by following its outbound path.

1.3.2. Objectives

To achieve the stated aim, the following objectives have been established:

1. **Optimize the Localization Pipeline:** Identify and implement the most effective pipeline, including parameter selection and methodological approaches, to achieve the highest possible localization accuracy while maintaining real-time performance and generalizability. This includes the selection, integration and optimization of feature extraction, matching, and planar transformation estimation techniques.

2. **Accurate Localization Estimation:** Estimate the latitude and longitude, using only prior telemetry data and image features, with a radial error below 10% of the UAV's radial displacement from the reference image. This accuracy is sufficient for manual control of the UAV in GPS-denied environments, given that the system provides continuous correction.

3. **Real-Time Operation:** Ensure the localization system operates in real-time, with a response time of less than 2 seconds following GNSS signal loss. This is sufficient as the relative motion of the UAV to the ground is sufficiently slow that a 2-second delay will not significantly affect the pilots ability to navigate.

4. **Environmental Generalizability:** Validate that the system maintains its performance metrics across diverse environments without the need for environment-specific parameter tuning.

OLD OLD

REQUIREMENTS

- **Accuracy:** Provide precise localization with a radial error below 10% of the UAV's radial displacement or movement from the centre of the reference image. This ensures sufficient accuracy for manual control to navigate the UAV along the outbound path.
- **Real-Time Operation:** Operate in real-time to support immediate navigation needs upon GNSS signal loss. This is defined as a response time of less than 2 seconds for position and heading estimation following GNSS signal loss. This ensures the navigator can make corrections prior to large changes in UAV position, given the landscape does not change significantly within 2 seconds.
- **Adaptability:** Maintain the Accuracy and Time constraints across diverse environments without requiring manually changing parameters per environment. This ensures the system's generalizability and applicability to various operational contexts.

OBJECTIVES

1. **Optimize the Localization Pipeline:** Identify and implement the most effective pipeline, including parameter selection and methodological approaches, to achieve the highest possible localization accuracy while maintaining real-time performance and generalizability. This includes the selection, integration and optimization of feature extraction, matching, and planar transformation estimation techniques.
2. **Accurate Localization Estimation:** Estimate the GPS location, using only prior telemetry data and image features, with a radial error below 10% of the UAV's radial displacement from the reference image. This accuracy is sufficient for manual control of the UAV in GPS-denied environments, given that the system provides continuous correction.
3. **Real-Time Operation:** Ensure the localization system operates in real-time, with a response time of less than 2 seconds following GPS signal loss. This is sufficient as the relative motion of the UAV to the ground is sufficiently slow that a 2-second delay will not significantly affect the pilots ability to navigate.
4. **Environmental Generalizability:** Validate that the system maintains its performance metrics across diverse environments without the need for environment-specific parameter tuning.

1. **Develop the Image-Based Navigation Pipeline:** Design and implement the components of the navigation system, including feature extraction, feature matching, and homography estimation techniques.
2. **Evaluate System Performance:** Test the system using real-world data to assess its accuracy, robustness, and real-time operation capabilities item
3. **Optimize for Real-Time Operation:** Enhance the efficiency of the system to ensure it meets real-time operational requirements, with response times suitable for UAV navigation.
4. **Validate Generalizability:**
5. **Validate Generalizability:** Test the system's performance under various environments to assess whether it challenging condtions, such as varying light levels and overlap conditions, to validate its robustness.
6. **Assess Practical Limitations:** Identify and address practical challenges such as environmental changes, dynamic objects, and visibility issues that may impact system performance.

– END OBJECTIVES –

1.4. Scope

The scope of this project is defined to ensure feasibility within the allocated timeframe and resources. The following assumptions and limitations outline the project's boundaries:

- **Camera Alignment and Environment:** The UAV's downward-facing camera is assumed to remain perfectly aligned downward throughout all operations, with no pitch or roll. The ground is assumed to be predominantly planar at the altitude of image capture, simplifying the model by reducing the complexity of perspective distortion correction.
- **Constant Altitude:** The UAV maintains a constant altitude during operations, negating the requirement for scale estimation in localization calculations.
- **Image Quality and Visibility:** Captured images are assumed to have adequate resolution and visibility, without significant distortion or occlusion.
- **Static Environment:** The UAV operates over primarily static land surfaces with minimal dynamic objects, ensuring a stable environment for image-based localization.

- **Path Found:** The system assumed that complementary odometry-based systems have located the UAV's path when GNSS signals are lost, and the overlap of reference image information is greater than 60%.
- **Methodology Focus:** The study aims to assess the system's viability using established methods and parameter sets, focusing on overall performance rather than exhaustive optimization of methods or parameters.
- **Machine Learning Constraints:** Machine learning methods requiring pre-training are excluded due to time and data constraints.
- **No Integration:** The system uses image-based data to output estimated position and heading information for the UAV's, intended for navigation assistance; it does not integrate practically with navigation systems or control mechanisms.

1.5. Data Provisioning

An agreement was established with an external party to provide real-world flight data from their UAV operations for use in this study. However, the data was not provided within the project's timeframe. To proceed, Google Earth data was utilized as it offers free access to high-resolution aerial imagery with 3D terrain features and perspective changes, along with GPS coordinates, closely approximating real-world data. The use of estimated heading data, which is required to convert translation changes to latitude and longitude, is acknowledged to have an impact on accuracy, but no other solutions were available that would more closely meet the project goals within the available resources.

1.6. Summary of Work

- MAY REMOVE This project successfully developed and implemented an image-based GPS localization system for UAV navigation in GPS-denied environments. The system met the outlined objectives, demonstrating high accuracy, real-time performance, and generalizability across multiple environments. Through comprehensive research and implementation of feature extraction, matching, and homographic techniques, the system effectively estimated the UAV's location with minimal error. The system was tested and deemed to be practically robust under low-light and limited overlap conditions, with the ability to handle various environments. The outcomes indicate significant potential for enhancing UAV navigation reliability in both military and civilian applications.

1.7. Structure of the Report

This report is structured to provide a comprehensive overview of the research undertaken to develop the image-based GPS localization system for UAVs. Chapter 2 reviews the current state of GPS-alternative UAV navigation systems and their vulnerabilities. Chapter 3 details the methodology, including feature extraction, matching, and homographic estimation. Chapter 4 describes the testing environment and datasets used, along with the comparative analysis of different methods. Chapter 5 presents the evaluation of the developed pipeline against the objectives. Chapter 6 summarizes the project's outcomes, evaluates the system's viability, and outlines recommendations for future work.

By addressing the critical need for a drift-free, cost-effective, and emission-free navigation solution, this project aims to enhance UAV operational resilience in environments likely to expereince GNSS-denial. The image-based navigation system developed will contribute to advancing UAV capabilities, ensuring mission success, and safeguarding assets and personnel.

Chapter 2

Literature Review

2.1. Related Work

Autonomous navigation for UAVs has been widely researched, with prominent methods like Simultaneous Localization and Mapping (SLAM) providing comprehensive mapping and localization. SLAM allows UAVs to construct detailed maps while tracking their position within it arafat2023vision. Typically, SLAM combines multiple sensors—such as cameras, LiDAR, and IMUs—to achieve accurate localization, making it particularly effective in indoor, repetitive environments like warehouses. However, building and maintaining a high-resolution map in SLAM is computationally intensive, consuming significant processing power and memory arafat2023vision. This makes it challenging for real-time applications in resource-limited UAVs. Additionally, SLAM is sensitive to map distortions and inaccuracies, which can degrade localization reliability arafat2023vision. Furthermore, UAV navigation generally requires only localized terrain data relevant to immediate surroundings, making SLAM excessive for UAV applications.

Optical flow is another technique used to estimate motion by tracking pixel shifts between frames, relying on the assumption of small, smooth movements [9]. This brightness constancy assumption fails with abrupt UAV movements, rotations, or scale changes, leading to unreliable motion estimates. Furthermore, optical flow is computationally intensive when calculating dense flow across entire frames, making it less suitable for real-time processing on resource-constrained UAVs [10].

Various studies specifically address UAV navigation through image-based solutions, leveraging feature matching techniques to estimate location.

The first study presents an image-based location estimation approach that combines relative positioning techniques through aerial image sequences to improve short-term odometry, enhancing UAV navigation reliability over short distances. This method emphasizes the importance of constant global correction but does not consider a solution that uses fixed reference images for navigation [11].

Another method, LoFTRS, uses deep learning-based image matching with semantic constraints to improve matching accuracy, providing refined feature correspondence that strengthens subsequent UAV localization tasks [12]. While this study offers valuable insights into potential pipelines, it lacks a detailed implementation and comparison of various methods across different terrains.

Ultimately, these methods do not provide a complete navigation pipeline, real-time adaptability, or testing across diverse, challenging, typical UAV conditions. Implementation details required for practical use in varied environments are also absent, limiting their proof of robustness in real-world scenarios.

This study addresses these gaps by presenting a comprehensive pipeline designed for UAVs, incorporating in-depth method comparisons, environmental adaptability, and rigorous testing under adverse conditions, making it a reliable and scalable framework for solving real-world UAV navigation challenges through image comparisons.

2.2. Background

This section provides a comprehensive overview of the fundamental concepts and techniques that underpin the proposed image-based UAV navigation system. By delving into feature extraction, matching, and planar transformations, it establishes the theoretical foundation essential for the subsequent system design and implementation. The discussion emphasizes the critical role of feature-based methods over direct approaches, setting the stage for understanding the chosen methodologies.

2.2.1. Feature Detectors

Feature extraction is a cornerstone of image-based UAV navigation, enabling the estimation of transformations such as rotation and translation between consecutive images. Feature detectors identify **keypoints**—distinct, repeatable points within an image—and generate **descriptors** that encapsulate information about the local image region surrounding each keypoint. These keypoints and descriptors are essential for accurate matching across multiple frames, allowing the system to track movement while maintaining invariance to changes in scale, rotation, and illumination. This precision is critical for accurately inferring both rotational and translational shifts between images. An example of feature detection and matching, to be explained in the subsequent section, is illustrated in Figure 2.1 as outputted by the navigation system. This image shows the top 50 matches between two images after rotational alignment.

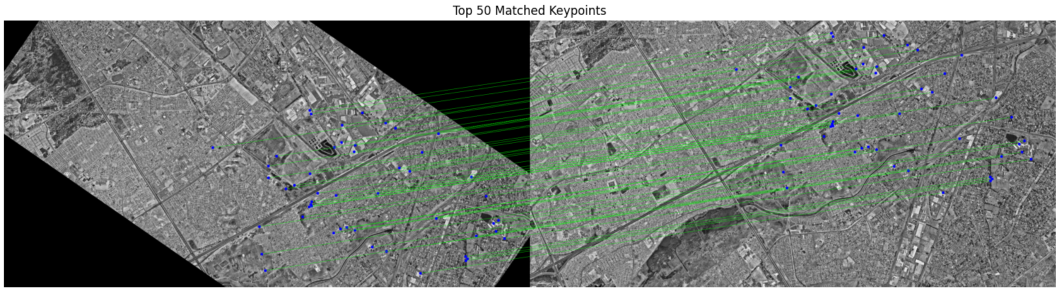


Figure 2.1: Feature Detection and Matching; adapted from [1].

In this context, *features* refer to both keypoints and descriptors, as each feature extraction method provides both components to facilitate effective image matching. The selection of feature extractors focuses on a subset of the most influential, efficient and accurate detectors that are free from patent restrictions.

ORB (Oriented FAST and Rotated BRIEF)

ORB combines the **FAST** [13] keypoint detector with the **BRIEF** [14] descriptor, enhanced for rotation invariance. **FAST** rapidly identifies keypoints by analyzing pixel intensity differences in a circular region around each candidate point. Once detected, **BRIEF** encodes the local image patch into a binary string through intensity comparisons. ORB introduces rotational invariance by aligning keypoints based on their dominant orientation before descriptor computation. This enhancement makes ORB both extremely fast and robust to scale and in-plane rotation, although it may struggle with repetitive textures or complex lighting variations [15].

AKAZE (Accelerated-KAZE)

AKAZE constructs a nonlinear scale space using diffusion-based filtering, capturing finer image details more effectively than linear methods. It detects keypoints by assessing local contrast with a specialized adaptive filter, enabling the identification of subtle features that simpler detectors might miss. The **Modified Local Difference Binary (MLDB)** [16] descriptor encodes the neighborhood of each keypoint into a binary vector based on pixel intensity differences. While AKAZE is both fast and compact, its performance can be sensitive to detection thresholds across different environments, potentially affecting its robustness in varied operational contexts [17]. However, a notable strength of AKAZE is its its rotational invariance [15].

SuperPoint with LightGlue

SuperPoint is a deep learning-based keypoint detector and descriptor that leverages convolutional neural networks (CNNs) to identify and describe keypoints in a single forward pass. Pre-trained on extensive image datasets, SuperPoint excels at recognizing stable and distinctive keypoints under varied conditions [18]. However, its performance may degrade on datasets significantly different from its training data. Pairing SuperPoint with **LightGlue**, a machine-learning-based matcher, enhances matching accuracy through advanced graph-based techniques that were recognized at the 2023 International Conference on Computer Vision [19]. Despite their high accuracy, SuperPoint and LightGlue are computationally intensive, necessitating GPU acceleration for real-time applications; however, their improved performance justifies their testing in this study, noting that they will not perform in real-time when testing with only a CPU [18].

2.2.2. Feature Matching

Feature matching establishes correspondences between keypoints in different images based on descriptor similarity. After identifying these correspondences, ambiguities and low-quality matches are removed, as detailed in Section 2.2.5.

Each matcher generates a list of potential matches along with their similarity scores, quantified using a descriptor-space distance metric. These scores are instrumental in subsequent filtering processes.

Feature matching involves two primary components: the choice of matching technique to acquire potential matches and the search technique that determines which of these matches to retain.

Types of Feature Matching Techniques

The following match acquisition techniques are commonly employed in feature-based navigation systems:

Brute-Force Matcher (BFMatcher): The Brute-Force Matcher is a simple, exhaustive matcher that matches each feature in one image with every feature in the second image, ensuring the best possible match based on descriptor similarity. While this guarantees high accuracy, it is computationally expensive, especially with large numbers of keypoints, making it less suitable for real-time applications without optimization [20].

Fast Library for Approximate Nearest Neighbours (FLANN): FLANN accelerates the nearest neighbour search in high-dimensional descriptor spaces using algorithms such as KD-trees or hierarchical clustering, adapting dynamically to the dataset. This approximate matching approach offers significant speed improvements with minimal loss in accuracy, making it ideal for real-time applications with extensive datasets [21].

LightGlue: Leveraging deep learning, LightGlue involves both a matching and search technique, and improves matching accuracy by employing advanced graph-based techniques to establish more reliable correspondences. Although highly effective, its structure necessitates the use of other neural network-based feature extractors like SuperPoint to realize its full potential [19]. The enhanced accuracy comes at the cost of increased computational demands, requiring GPU acceleration for optimal performance.

The following search techniques are commonly used to filter matches and retain only the most reliable correspondences:

Radius Search: This method retains matches within a specified distance in descriptor space, effectively filtering out weaker matches. However, it does not guarantee a fixed number of matches per keypoint, leading to inconsistent results [22].

K-Nearest Neighbours (KNN) Matching: KNN matching retains the top K matches for each keypoint, allowing the application of post-filtering techniques such as Lowe's ratio test to eliminate ambiguous matches [22].

Vanilla Matching: Vanilla matching returns the single best match for each keypoint based on the closest descriptor distance. It is a subset of KNN matching with K=1, offering simplicity and ease of implementation [22].

2.2.3. Image Similarity Computation

Image similarity computation is a pivotal component of UAV navigation systems that rely on reference images for accurate localization and pose estimation. Effective similarity measures ensure efficient processing of extensive image datasets and facilitate precise transformation estimations, which are essential for reliable navigation.

2.2.3.1 Proximity-Based Techniques

To achieve efficient, real-time performance, the search space is reduced to images within the proximity of UAV's last known location, filtering images within a static or dynamic radius. While this method is highly efficient, it does not account for potential deviations from the expected flight path or the presence of poor-quality reference images. This limitation implicates that this measure cannot be used as the sole basis for image similarity computation, necessitating the integration of additional techniques for comprehensive assessment.

2.2.3.2 Global Matching Techniques

To ensure images are evaluated for similarity and ensure they are free from significant distortion, global matching, or direct methods are employed. Direct methods estimate planar transformations by comparing entire image pixel intensities and minimizing differences through optimization techniques like gradient descent. Due to their entire context, they are well-suited for similarity comparisons, but they are not ideal for precise transformation estimation due to their sensitivity to noise and illumination changes [23]. The following methods are global matching techniques commonly used in image similarity computation:

Cross-Correlation

Cross-correlation measures similarity by sliding one image over another and computing the sum of pixel-wise multiplications at each position. The peak value signifies the best alignment, and its magnitude indicates the confidence level of the similarity. Higher confidence values reflect greater similarity between the images. While straightforward to implement, cross-correlation is sensitive to noise and illumination changes, which can compromise the reliability of the similarity measure [24].

Histograms

Histogram comparison assesses similarity by analyzing the distribution of pixel intensities within each image. Typically, each image's histogram is divided into 256 intensity bins for 8-bit images, and similarity is quantified using metrics such as Chi-Square or Bhattacharyya distance. This method emphasizes global color and brightness distributions but neglects spatial information, making it less effective for nuanced structural differences [25].

Structural Similarity Index (SSIM)

SSIM evaluates similarity by decomposing images into luminance, contrast, and structure components. It computes local statistics within small windows and integrates them into a single similarity score that mirrors perceived image quality. SSIM effectively captures structural information like edges and textures, aligning closely with human visual perception. Although slightly more computationally expensive than the former methods, it is robust to varied conditions [26].

Local Detectors Conversion Although not inherently a global matching technique, local feature matching can be adapted to achieve a global understanding of image similarity. This involves identifying and matching keypoints in both images and assessing the overall number of good matches. However, this approach alone does not ensure an even distribution of matches across the entire image, potentially leading to biased, localized similarity assessments. To mitigate this, a grid matching technique is employed, dividing the image into grids and limiting the number of matches per grid. Although the most computationally intensive, this method enhances robustness against distortions and rotations by ensuring a uniform distribution of matches across the image. To maintain reasonable runtime, this method has to employ a very crude detection and matching layer.

2.2.4. Planar Transformation Estimators

Feature-based methods extract and match keypoints from both reference and real-time images, often incorporating outlier removal stages [23]. By focusing on distinctive features rather than every pixel, these methods excel in handling large viewpoint changes and rotations, enabling more precise transformation inference. This targeted approach enhances computational efficiency and robustness to environmental distortions, though it requires careful management to avoid performance degradation from variation in the number of extracted feature points.

In typical UAV flight scenarios, the primary transformations of interest are rotation and translation, as perspective distortion, caused by 3-Dimensional structures, and shear distortion, caused when the UAV turns or generally is not parallel to the ground, are minimal at high altitudes and while the UAV is level respectively. Further, scaling is not present as per scope. The following subsections detail the primary planar transformations employed in the system.

Affine Transformation

Affine transformation captures translation, rotation, scaling, and shear, providing six degrees of freedom. It is represented by a 2×3 matrix that maps points from one plane to another while preserving lines and parallelism. Affine transformations are computed by estimating the affine transformation matrix between two sets of corresponding points using OpenCV's `estimateAffine2D` function [27]. While versatile, the inclusion of scaling and shear introduce unnecessary error points in the UAV case.

Rigid Transformation Estimation (SVD)

The rigid transformation via SVD preserves the shape and size of objects by estimating only rotation and translation, excluding scaling and shear. Represented by a 2×3 matrix, rigid transformation ensures orthogonality in the rotation component. Utilizing Singular Value Decomposition (SVD), this method minimizes the least-squares error between two point sets. The process involves: computing the weighted centroids of both point sets, centering the points by subtracting their respective centroids, calculating the covariance matrix of the centered points, performing SVD on the covariance matrix to derive the rotation matrix, and determining the translation vector based on the centroids. The resulting 2×2 rotation matrix and 2×1 translation vector are combined to form the rigid transformation matrix. Rigid transformation is computationally efficient and well-suited for UAV applications [28].

Partial Affine Transformation

Partial affine transformation simplifies the full affine model by focusing solely on translation, rotation, and limited uniform scaling, offering four degrees of freedom. This transformation is also represented by a 2×3 matrix, similar to the affine transformation but without shearing and with reduced, uniform scaling. This offers similar performance to the prior rigid method, but with minor additional error points due to scaling [27].

Homography Transformation

Homography transformation accounts for translation, rotation, scaling, shear, and perspective distortion, providing eight degrees of freedom. It is represented by a 2×3 matrix and is estimated using OpenCV's `findHomography` function, typically with RANSAC for outlier rejection [29]. While homography offers greater flexibility in modeling complex, typically 3-Dimensional transformations, its additional degrees of freedom introduce unnecessary errors and computational overhead for UAV-based applications.

2.2.5. Optimization Techniques

Optimization techniques aim to refine the accuracy and reliability of the matched points used for transformation estimation by effectively filtering out erroneous matches and improving transformation accuracy. The following subsections detail the primary optimization methods employed in this system.

Random Sample Consensus (RANSAC) for Planar Transformation

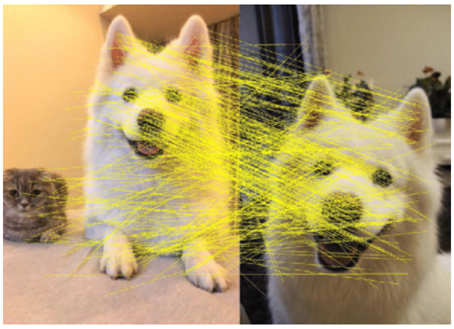
RANSAC is a robust estimation technique used to estimate planar transformations by iteratively selecting random subsets of point correspondences to fit a model and identify inliers [30]. The process involves randomly selecting a minimal subset of point pairs, estimating the transformation model (e.g., affine or homography) based on the selected subset, determining the number of inliers that fit the estimated model within a predefined threshold, and repeating the process for a set number of iterations or until a sufficient inlier ratio is achieved. This approach is highly effective in datasets with significant outliers, focusing on finding a model that best fits the largest subset of inliers. However, due to its iterative nature and the need to sample repeatedly, RANSAC can result in increased runtime, particularly in larger datasets or when dealing with numerous outliers [30].

Local Maxima Extrema Density Selection (LMEDS) for Planar Transformation

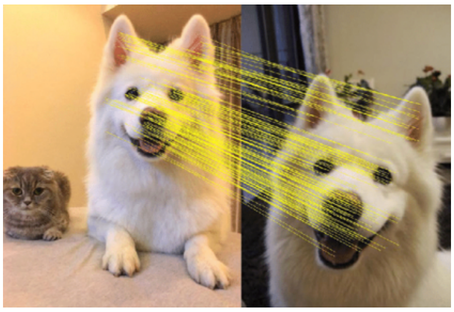
LMEDS is a keypoint selection method designed to prioritize areas of high feature density by identifying local maxima as keypoints for matching [31]. This technique involves analyzing the image to identify regions with high feature density, selecting keypoints located at local maxima within these dense regions, and filtering out less significant keypoints to reduce redundancy and improve match quality. By concentrating on areas with high feature concentration, LMEDS ensures that keypoints represent the most distinctive and informative regions of the image, enhancing both accuracy and performance by reducing redundant keypoints and improving match quality, particularly in areas with high feature variability.

Lowe's Ratio Test

Lowe's ratio test is a widely used filtering technique used to eliminate ambiguous or false keypoint matches by comparing the distance of the best match to the second-best match [2]. For each keypoint match, the ratio of the distance of the best match to that of the second-best match is calculated, and the match is retained if this ratio is below a predefined threshold. A lower ratio indicates that the best match is significantly better than the alternatives, thereby increasing the likelihood of the match being correct. An example of Lowe's ratio test filtration are showed in figure 2.2a and figure 2.2b



(a) Matching Features Without Lowe's Ratio Test.



(b) Matching Features With Lowe's Test.

Figure 2.2: Reproduced from [2].

N-Match or Absolute Thresholding

N-match thresholding involves setting a threshold that allows only a specific number of matches with the smallest descriptor distances to be retained. Absolute thresholding filters matches based on a fixed distance in descriptor space. Only matches that meet or fall below this predefined distance threshold are retained, ensuring that only sufficiently similar matches are used in the transformation estimation process. These methods primarily suffer from difficulty in setting the threshold, which is often extremely sensitive to dataset variations and method parameters.

Chapter 3

System Design

This chapter outlines the chosen methodology for the UAV navigation system, detailing the pipeline and its various components. The system is designed to accurately estimate the UAV's position and heading, even in scenarios where GPS data is unreliable or unavailable. The high-level flow of the system is illustrated in Figure 3.1.

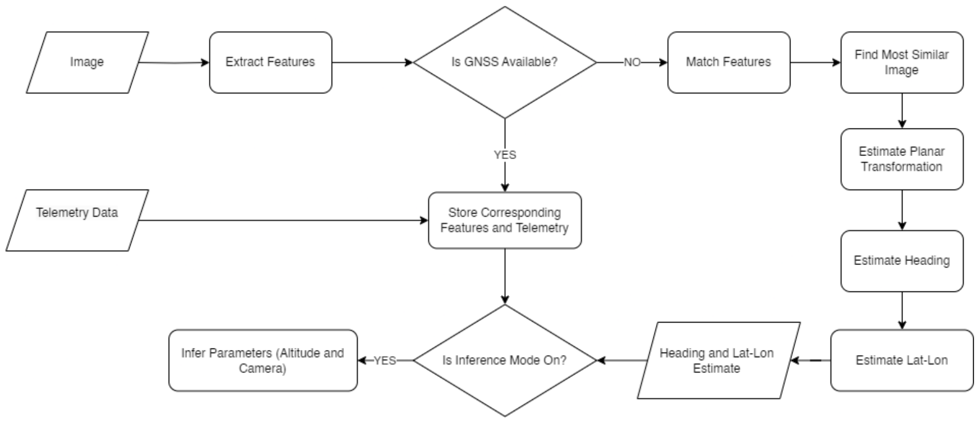


Figure 3.1: High-Level Flow of the System

3.1. System Pipeline

The UAV image-processing pipeline comprises several stages, each addressing specific goals in the larger aim to estimate the current GPS and heading of the UAV. These stages require varying levels of precision and efficiency to ensure robust performance.

3.1.1. Pipeline Stages

1. **Image (Input):** The process begins with capturing a live image from the UAV's downward-facing camera. This real-time visual input provides essential information about the UAV's environment, forming the basis for position estimation.

2. **Extract and Store Features:** Keypoints and descriptors are extracted from the current image to aid in the matching process. This extraction occurs in two layers: the coarse layer, used in the second stage of search space reduction, and the dense layer, used during the precise transformation stage. The system also extracts features

when GPS is available to reduce computational load during critical phases when GNSS is lost.

3. **Store Corresponding Features and Telemetry:** Extracted features (keypoints and descriptors) along with their corresponding telemetry data (GNSS position and heading) are stored for future reference. Stored features facilitate relative transformation inference when GNSS is unavailable, while telemetry data assists in converting relative transformations to real-world coordinates and headings.

4. **Infer Parameters (Altitude and Camera):** To be able to convert from a pixel translation to a metre value, a conversion factor is required. This conversion factor is dependent on the altitude of the UAV and the camera's focal length. When GNSS signal is available, this stage uses the complete pipeline to estimate the UAV's Lat-Lon coordinates using a placeholder factor value. These estimates are accumulated and compared, using only the first 5 images to balance overall efficiency and parameter accuracy, with the ground truth Lat-Lon coordinates via linear regression to infer the conversion factor. This is necessary in the scope of this study, which assumes constant and unknown altitude and camera parameters. These parameters are both represented within a single factor, the pixel to metre ratio.

5. **Match Features:** Features between the current image and reference images are matched to ensure that comparisons are based on mutual features. This involves matching of both the coarse and dense layers of features.

   The matching process is optimized to ensure robustness against noise and outliers, with a focus on computational efficiency. The techniques include usage of match acquisition techniques, search techniques, as well as subsequent optimization techniques to refine the matches.

6. **Find Most Similar Image:** The stage involves two layers of search space reduction until a single image is chosen for the current image to accurately infer its relative transformation against.

   The first is reduction based on proximity to the last known Lat-lon Coordinates. This stage outputs 5 matches.

   Thereafter, a more precise global matching technique is applied to identify the most similar match from the reduced search space. This requires initial rotational alignment using the coarse layer of matched features and subsequent global matching techniques to identify the best match based on similarity scores. The crude layer is chosen as global matching techniques were proven in the testing phase to be robust against minor rotational inaccuracies, allowing for efficiency prioritization.

7. **Estimate Planar Transformation:** After identifying the best match, the system performs a precise estimation of both rotation and translation between the input and reference images using the dense match layer.

   The first step involves estimating the rotation between the images, followed by aligning the images based on this rotation.

   Thereafter, the system recomputes the dense layer of features and matches on the aligned images. The reason for the recomputation prior to translation estimation is due to improved accuracy; aligned point clouds allow for improved translation estimation by way of less parameters to estimate. The amount of mutual information also remains constant when applying alignment to an image.

   Finally, the system estimates the translation between the images using the refined dense layer.

   This rotation and translation estimation are outputted to the next stages for conversion from relative to absolute conversion of the UAV's heading and position.

8. **Estimate Heading:** The internal angle between the current and reference images is added to the reference image's heading to determine the UAV's current heading.

9. **Estimate Lat-Lon:**

   The estimated translation at this stage is a pixel value representing the displacement between aligned images. To convert this to real-world latitude and longitude coordinates, the vector must be scaled in magnitude and rotated to align with the global coordinate system. This process consists of the following steps:

   First, the translation vector components, representing displacement between the current and reference images, are initially relative to the internal coordinate system. This vector is rotated by the heading of the reference image, aligning it with the global latitude-longitude system without altering its magnitude. *At this stage, the translation vector is aligned with the global coordinate system but remains in pixel units.*

   Next, the estimated pixel displacement is multiplied by the inferred pixel-to-metre conversion factor, resulting in a translation vector scaled to represent real-world distances in metres. *At this stage, the translation vector represents real-world distances in metres and is aligned with the global coordinate system.*

   Then, the metre-based displacement components, now aligned with the global coordinate system, are converted to relative latitude and longitude changes. To account for the Earth's oblate spheroid shape, the equations below are used, which consider the latitude-dependent distance per degree of longitude:

$$\Delta \text{Longitude} = \frac{\Delta \text{East}_{\text{metres}}}{111320 \cdot \cos(\text{Latitude}_{\text{reference}})}, \quad \Delta \text{Latitude} = \frac{\Delta \text{North}_{\text{metres}}}{111320}$$

where:

- $\Delta\text{East}_{\text{metres}}$ and $\Delta\text{North}_{\text{metres}}$ represent the eastward and northward displacement vector components in metres, aligned to the global coordinate system.
- The constant 111320 converts degrees of latitude to metres, with longitude scaled by $\cos(\text{Latitude}_{\text{reference}})$ to reflect latitude-dependent longitudinal distance.

At this stage, the translation vector represents relative changes in latitude and longitude, aligned with the global coordinate system.

Finally, the calculated changes in latitude and longitude are added to the reference image's known coordinates, yielding the UAV's estimated absolute position.

10. **Heading and Lat-Lon Estimate:** The systems heading and Lat-Lon estimates are outputted to the user interface for real-time monitoring and in practice, navigation.

3.2. Dynamic Methods and Techniques

Upon rigorous testing, it was seen that not all parameters in the pipeline can generalize without tuning. However, static environmental tuning is not always possible or practical. To get around this, two dynamic methods were implemented to adjust these parameters to better account for varying densities of quality and quantities of keypoints across datasets.

Firstly, the keypoint filtering threshold of AKAZE was inconsistent across datasets, with static thresholds subtending unreasonably low or high numbers keypoints in different datasets. To address this, a dynamic method was implemented to adjust the leniency threshold based on the number of keypoints detected. This is done for the first image in the dataset, since changing terrains were not considered in this study. This ensures a stable number of keypoints were found. Specifically, enough to ensure a stable estimate but not too many to cause excessive computational overheads and noise.

Secondly, Lowe's ratio, an extremely powerful filtering technique in the matching process, was found to be highly sensitive to the dataset. This was caused by varying keypoint qualities for the same number of keypoints across datasets. To address this, a dynamically adjusting threshold was applied. This threshold increases in leniency until a sufficient number of matches are found or a certain percentage of the total number of keypoints are matched.

The above dynamic methods ensure that the system remains stable in different scenarios, where prior knowledge of the environment is not available. Naturally, this may be adapted to re-estimate these parameters, based on a time constraint, in varying environment missions.

3.3. Testing Shortlist

The system design integrates multiple features to ensure robustness, accuracy, and efficiency in UAV navigation tasks. The following components have been selected and tested across various scenarios:

- **Feature Detectors:** AKAZE, SuperPoint (with LightGlue matcher), and ORB. Note, the SuperPoint detector is used in conjunction with the LightGlue matcher to enhance performance.
- **Feature Matchers:** FLANN and BruteForce. KNN
- **Search Techniques:** KNN with $K = 2$. Empirical tests showed a value of 1 to be ineffective without Lowe's ratio test, while values above 2 introduced excessive computational overheads. Further, radius search was seen to be unreliable due to the varying density of keypoints across datasets.
- **Planar Transformation Estimation:** Homography, Affine, Partial Affine (Rigid) transformations using OpenCV, and SVD-based rigid transformations for rotation and translation estimations. Rotational and Translational estimates were initially tested separately, due to the possibility of different responses to different prior stages and methods. However, it was seen that inter-method comparisons subtended equivalent conclusions; the methods were tested for brevity using a combined transform.
- **Global Image Similarity Measures:** SSIM, histogram matching, local retrofit, and cross-correlation.
- **Optimization Techniques:** Standard Deviation Filtering, LMEDS, RANSAC, Lowe's ratio test, n-Match thresholding, and absolute thresholding for match refinement. Empirical tests indicated that cross-checking was not applicable due to the excessive computational costs involved. KNN

3.4. Conclusion

The system design integrates feature extraction, matching, and transformation estimation to enable precise UAV navigation, even in the absence of reliable GPS data. By employing a combination of diverse feature detectors and optimizing various pipeline stages, the system ensures robustness, accuracy, and computational efficiency across different operational scenarios.

Chapter 4

Comparison of Methods

This chapter compares the performance of various methods used in the UAV navigation system, focusing on accuracy, runtime, and robustness. The evaluation includes feature detectors, local feature matchers, rotational and translational estimators, and optimization techniques. Each method is tested across multiple datasets to assess generalization and suitability for real-world UAV applications.

4.1. Testing Setup

This section outlines the framework used to evaluate the performance of the proposed UAV navigation methods. The evaluation focuses on three primary metrics: accuracy, runtime, and robustness. These metrics are critical to ensure that the navigation system can reliably operate under diverse and challenging conditions, reflecting real-world scenarios where the UAV may encounter varying environmental factors.

Note that the testing setup is largely reused, and not reexplained, in the results chapter.

4.1.1. Critical Testing Pipeline Understanding

The aim of these tests were to compare different methods and not to evaluate the system as a whole. These tests were not tested under optimal runtime or accuracy settings; however, other stages and parameters are held constant when testing between methods in a given stage. Further, sufficiently optimal non-testing parameters and methods were chosen such that the methods are tested under their intended optimal conditions. The results are not indicative of the overall performance of the system, and no comments thereon are made in this chapter. Specifically, no objective conclusions may be made about the overall performance in this chapter.

Secondly, to ensure maximal usage of the dataset, the same images are used carefully in both the GPS and no GPS stages to ensure no bias. In the GPS Available phase, all 15 images are streamed and added, storing their features and telemetry. Further, the first 5 images are used to infer the fixed pixel to metre factor, related to the camera focal length and UAV altitude. Thereafter, in the no GPS phase, the last 14 images have their location and heading estimate. Naturally, each image recomputes its features, and does not infer its location based on its own image from the GPS pipeline, to ensure a fair test.

4.1.2. Datasets

Five distinct datasets were selected to rigorously evaluate the methods' generalization and performance across diverse environments. These datasets were captured using Google Earth and involved both translational and rotational movements, simulating typical UAV navigation tasks. Although the primary transformations were translation and rotation, perspective distortions and other subtle deformations were present due to the uneven terrain. Multiple methods with varying degrees of freedom were tested to implicitly estimate which of these transforms were significantly present. Each of the properties and choices were done to ensure the task was challenging but not impossible.

General Properties of the Datasets:

- **Altitude:** Ranging from 5.5 to 6.5 km, constant within each dataset. This is standard for UAV navigation applications and ensures the ground is sufficiently planar.
- **Resolution:** 1920x972 pixels per image. This is the standard resolution in most applications. Stress tests were performed at lower resolutions.
- **Radial Movement Between Frames:** Varies between 100 and 700 pixels depending on the image and the best match found, as is the case in reality. This variability does not impact inter-method comparisons. Metres are used for physical clarity, where, despite differences in scaling, a lower metre error directly indicates a lower percentage error between methods.
- **Number of Images per Dataset:** 15 images per dataset. This balances testing time and is sufficient to evaluate the methods' performance.

The datasets used for testing were as follows:

- **CITY1 and CITY2 (Cape Town):** Both datasets were captured in Cape Town. **CITY1** includes both rotational and translational changes between frames, while **CITY2** is the only dataset that only focuses on translational movements. This distinction allows for isolated testing of performance under rotational stress.
- **ROCKY:** This dataset, taken in the semi-arid Karoo region, involved rugged terrain with sparse vegetation. They involve translations and rotations.
- **DESERT and AMAZON:** The desert dataset was captured in the Sahara Desert, while the Amazon dataset was captured in the Amazon Rainforest. These datasets

are characterized by sparse, repetitive patterns, making it challenging even for human observers to distinguish differences between frames. They present significant difficulty for feature extraction and matching. They both involve translations and rotations.

Examples of the datasets are shown below:

Figure 4.1: Examples of the CITY1 and CITY2 Datasets.

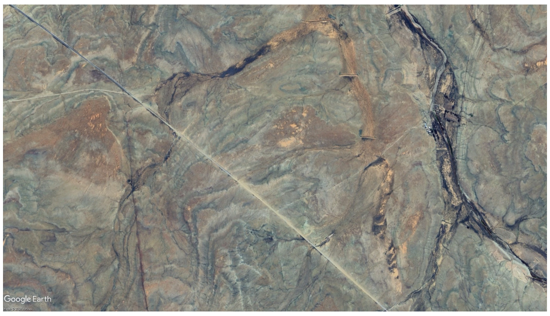


Figure 4.2: Example of the ROCKY Dataset.

Figure 4.3: Example of the DESERT Dataset.

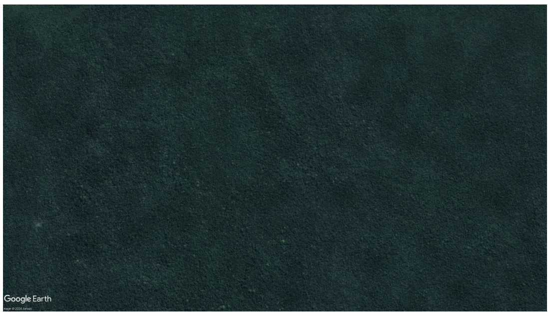


Figure 4.4: Example of the AMAZON Dataset.

Figure 4.5: Overview of Datasets

4.1.3. Testing Structure

Each method is subjected to rigorous testing based on the following criteria:

- **Accuracy**: Evaluated using the radial Root Mean Square Error (RMSE) of GPS estimations, in metres. This metric is chosen, instead of percentage error, for a physical interpretation of the error, providing a clear indication of the method's accuracy in estimating the UAV's position. This does not compromise the inter-method comparisons, as the same metric is used across all methods. Further, this accuracy is only tested at the end of the pipeline, since any error or poor choice in prior stages will always propagate to the final GPS error; and often, the accuracy and impact of intermediate stages are not directly interpretable. This radial error is given as the mean of the radial errors for all images in the dataset.
- **Runtime**: The runtime of the entire dataset is used as the method of comparison. This implies the runtime to compute all 15 images, for both the GPS and no GPS

phases. As before, intermediary stages propagate this error, and the runtime per line is not necessarily indicative of better performance; Runtime per line is not tested. For instance, a method may quickly compute filtering, but if it does not filter much, later stages will be slower. slower.

- **Robustness**: This test specifically employs qualitative testing to evaluate the method's performance under variation in its parameters. That is, how hard is this method to tune, and how does it perform with crudely chosen parameters across datasets. This aims to evaluate the method's generalizability and robustness across diverse environments. A scoring system is developed to balance the nuances of parameter sensitivity evaluation by considering the range of parameters that can be chosen, and the performance of that range across datasets. Specifically, the scoring system, from 1-5, is as follows:
  - 5 indicates a wide range of parameters offer near perfect (defined as the performance of the most optimal parameter) performance across datasets,
  - 4 indicates a large range of parameters offer good performance across datasets or a small range offers near perfect performance,
  - 3 indicates a small range of parameters offer good performance across datasets,
  - 2 indicates a small range of parameters offer significantly performance across datasets, and
  - 1 indicates that no static threshold will work across datasets.

This structured testing approach ensures that each component of the UAV navigation system is thoroughly evaluated, facilitating the selection of methods that deliver optimal performance across all critical metrics.

4.2. Feature Detectors

This section presents the evaluation results of three feature detectors: ORB, AKAZE, and SuperPoint with LightGlue. Feature detectors were applied to extracting a crude feature layer for rotational estimation, as well as a dense layer for rotational and translational estimation. Initially, all 3 transformations were tested independently for each detector, however, it was seen that the inter-method comparison conclusions were equivalent for each stage. Therefore, to maintain brevity, the detectors are applied to all stages and compared once, with the understanding of equivalent comparative independent responses to each stage.

4.2.1. Accuracy and Runtime

Figure 4.6 and Figure 4.7 present the radial error and runtime values for each feature detector across different datasets. AKAZE, utilizing dynamic keypoint targeting, demonstrated the highest accuracy across all datasets while maintaining reasonable runtime. SuperPoint recorded the highest radial errors, particularly in challenging datasets such as ROCKY and AMAZON, indicating its limited generalizability across diverse environments due to its training set. Meanwhile, ORB proved to be the most efficient detector, making it suitable for applications requiring fast processing. SuperPoint demonstrated the longest runtimes across all datasets, highlighting its limited applicability for time-sensitive applications unless optimized with GPU acceleration.

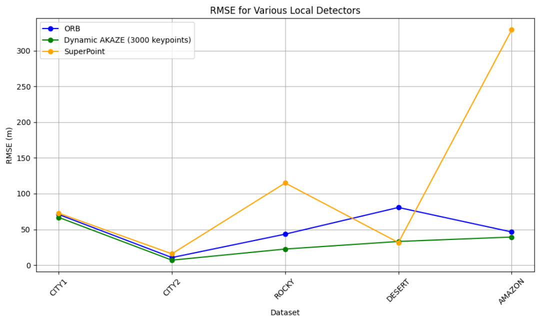


Figure 4.6: Radial Error for Various Local Detectors.

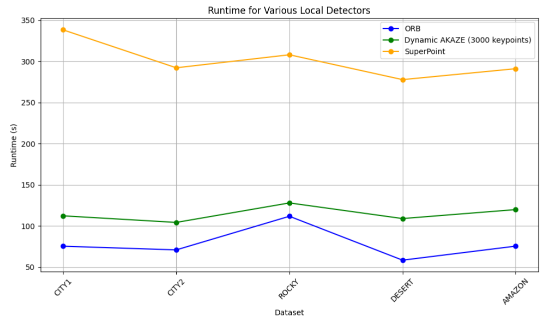


Figure 4.7: Runtime for Various Local Detectors.

Figure 4.8: RMSE and Runtime Comparison for ORB, AKAZE, and SuperPoint Across Datasets.

Robustness

All the 3 methods were able to maintain their keypoint targets across environments; AKAZE is evaluated in terms of its dynamic keypoint targeting implementation. However, the quality of keypoints varied significantly across datasets, leading to variant performances, showing the downsides of static keypoint targeting. SuperPoint consistently achieved good performance across multiple keypoint targets, achieving it a robustness score of 5. AKAZE, with its dynamic targeting, was able to generalize well across datasets and only significantly dropped in performance when using below 1000 keypoints, achieving a robustness score of 4. ORB, while efficient, required precise tuning to achieve consistently good performance across datasets and was highly sensitive to the target parameter, achieving a robustness score of 3.

4.2.2. Final Selection of Feature Detectors

Based on the comprehensive evaluation, the following detectors were selected for the respective stages of the UAV navigation system:

- **Crude Layer (Initial Detection)**: **ORB** was chosen for the crude, rotational estimation layer due to its balance of accuracy and efficiency. A range of 3000 - 8000 keypoints maintained reasonable accuracy and runtime, largely due to the invariance of the image similarity estimators to rotational inaccuracies. 3000 keypoints was chosen to prioritize speed and maintain FLANN runtime as per 4.3.3.
- **Dense Layer (Refined Detection)**: **Dynamic AKAZE** was selected for the dense layer due to its consistent performance and robustness and applied to both rotational and translational estimation stages. A range of 3000-5000 maintained consistent runtime and accuracy. 3000 keypoints was chosen to balance runtime, accuracy and maintain FLANN runtime as per 4.3.3.

4.3. Local Feature Matchers

This section evaluates two prominent local matchers, BFMatcher and FLANN, within the context of a UAV navigation system. Note that LightGLue was implicitly tested in the feature detectors section with SuperPoint, as explained in the prior two sections.

4.3.1. Accuracy and Runtime Evaluation

Figure 4.9 presents the radial error in GPS values for BFMatcher and FLANN across different datasets. The results indicate that while BFMatcher achieves slightly better accuracy in certain cases, FLANN remains highly competitive with only marginally higher RMSE values.

Figure 4.10 shows the runtime comparison for BFMatcher and FLANN across different datasets. FLANN consistently outperforms BFMatcher in terms of speed, with significantly lower execution times across all datasets.Specifically, in the CITY datasets, where the number of keypoints found were significantly higher, despite keypoint targeting, BF-Matcher's runtime was significantly higher than FLANN's. This is attributed to FLANN's approximate matching, which scales better with the number of keypoints.

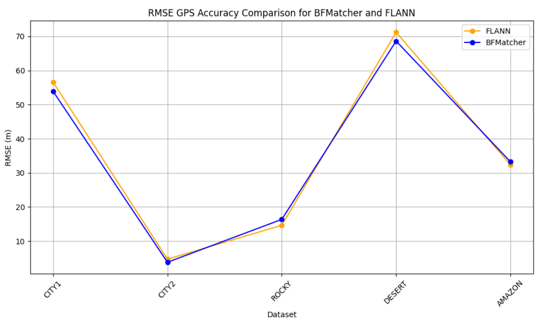


Figure 4.9: Radial Error for BF-Matcher and FLANN.

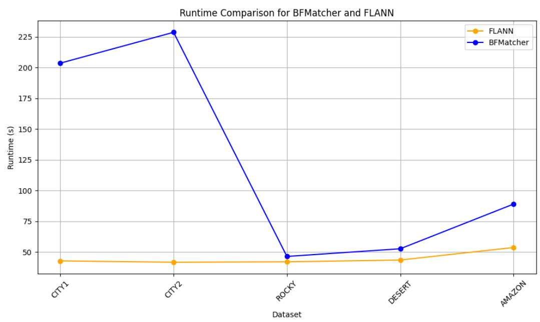


Figure 4.10: Runtime Comparison for BFMatcher and FLANN.

4.3.2. Robustness Testing

BFMatcher did not have any tunable parameters. FLANN did, however, it showed limited variation in accuracy across the entire spectrum of parameters. As such, both methods achieve a robustness score of 5.

4.3.3. Considerations

This section details the scenarios in which each method should be employed. Figure 4.11 depicts the convergence in RMSE GPS error between FLANN and BFMatcher as the number of keypoints increases. Positive values indicate that BFMatcher outperforms FLANN, while negative values suggest FLANN performs better. The plot demonstrates the convergence in accuracy of both matchers as the number of keypoints increases, with FLANN generally maintaining slightly higher RMSE values than BFMatcher. Note that outliers are present, indicating instances where FLANN achieves better accuracy than BFMatcher. This is attributed to FLANN's approximate matching, paired with a static Lowe's ratio threshold, which may preserve valid matches that BFMatcher might discard due to finding a better second match. Notably, if FLANN is to be employed, at least 3000 keypoints should be targeted to ensure consistent performance across datasets. If significantly below this, BF matcher should be considered.

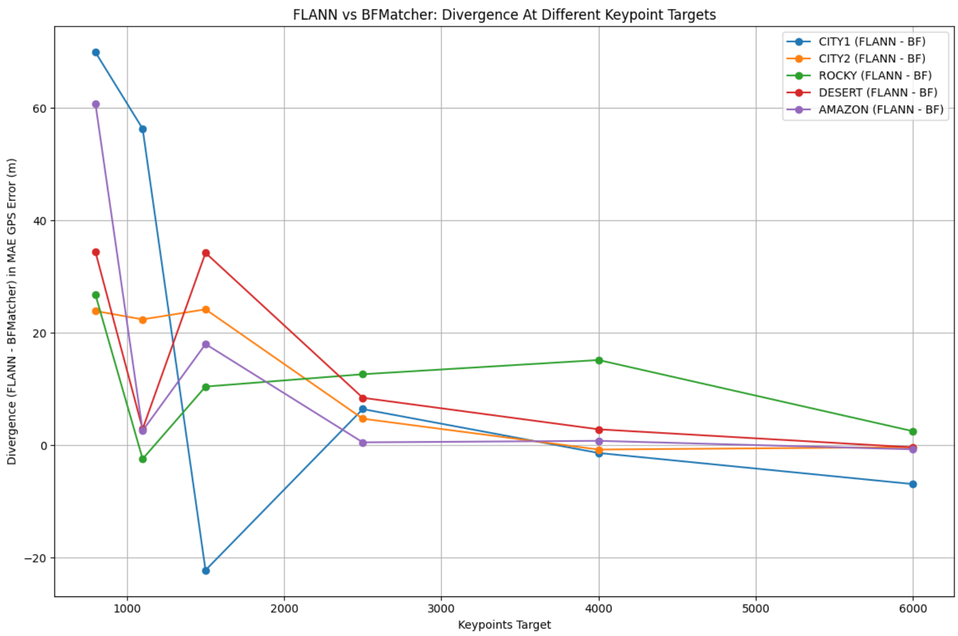


Figure 4.11: Convergence in RMSE GPS Error Between FLANN and BFMatcher Across Keypoint Targets.

4.3.4. Final Selection of Local Feature Matcher

Based on the comprehensive evaluation of accuracy, runtime, and robustness, FLANN emerges as the optimal choice for the UAV navigation system. FLANN offers significantly faster runtimes and better scalability while maintaining comparable accuracy to BFMatcher. However, at least 3000 keypoints should be targeted to ensure consistent performance across datasets.

4.4. Planar Transform Estimators

This section evaluates the performance of four planar transformation estimation methods: Partial Affine 2D (Rigid Transform plus minor Scaling), Affine 2D, Homography, and Rigid Transform Via SVD. As noted in section 3.3, both rotational and translational stages are combined into a single transform evaluation due to conclusion equivalency. For these tests, RANSAC was used on all methods, except the Rigid Transform Via SVD which does not have the method built-in, to filter outliers.

4.4.1. Accuracy and Runtime Evaluation

Figure 4.14 summarizes the RMSE and runtime values across datasets for each rotational estimator method. From the results, it is clear that each method exhibits strengths in different datasets. However, the Rigid SVD method and Partial Affine 2D consistently emerge as the most accurate methods. This implies that the Rigid SVD transform provides the most stable and accurate estimate of the rotational and translational components of the UAV's movement. The Rigid SVD method slightly outperforms Partial Affine 2D in most datasets, attributed to its strict rigid transformation alignment with dataset characteristics.

The runtime results, shown in 4.14, are influenced by the degrees of freedom in each method, with rigid-based transforms demonstrating the best performance across datasets. SVD proves the fastest due to its lack of iterative optimization.

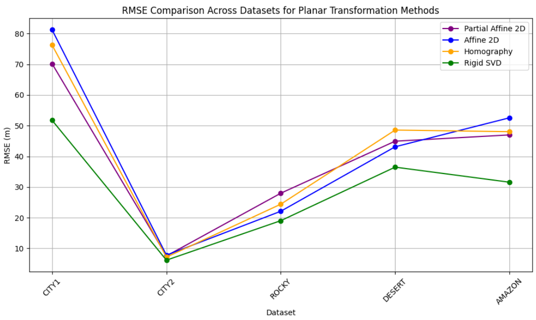


Figure 4.12: RMSE Comparison Across Datasets for Planar Transform Estimators.

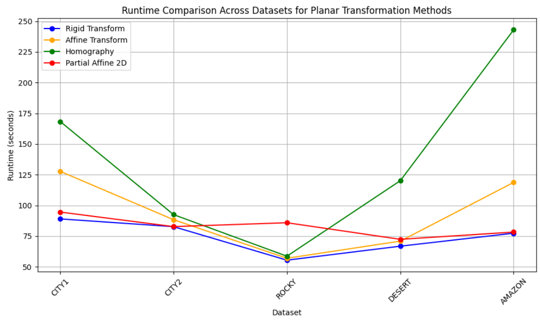


Figure 4.13: Runtime Comparison Across Datasets for Planar Transform Estimators.

Figure 4.14: RMSE and Runtime Comparison Across Datasets for Planar Transform Estimators.

4.4.2. Robustness Testing

The openCV methods (partial Affine 2D, Affine 2D, homography) utilize RASNAC for robust outlier rejection. However, the RANSAC threshold subtended large differences in accuracy when varied, and required threshold testing from very low to very high thresholds to find the optimal threshold. For more complex, 3-Dimensional, transformations this threshold may be more useful, but it was found to not beat the accuracy of the Rigid SVD method under any threshold in the given application. The OpenCV methods achieve a score of 2 for robustness, while the Rigid SVD method achieves a score of 5 due to its lack of tunable parameters.

4.4.3. Final Selection of Rotational Estimator

Based on the comprehensive evaluation of accuracy, runtime, and robustness, the rigid transform by SVD emerged as the most suitable estimator for the UAV navigation system. It demonstrated the lowest combined radial RMSE in GPS across all datasets, and the fastest runtime, largely due to its application-aligned degrees of freedom. Further, it required no parameter tuning, making it highly suitable for real-time UAV applications.

4.5. Image Similarity Estimators

Accurate image similarity estimation, or global matching, is essential for UAV navigation systems to choose reasonable images to compare to. They should ensure accuracy and efficiency while maintaining robustness against small rotational offsets. The proximity radius for initial search space reduction was crudely set to 5, but this is not tested as it is a design choice dependent on external factors such as the UAV's speed and the image capture rate. This section specifically evaluates the global matching techniques which are used to estimate the similarity between images to infer the closest match for subsequent heading and position estimation.

4.5.1. Accuracy and Runtime Evaluation

For clarity, the best match found is implicitly realized in the GPS estimation error because more similar matches will have more accurate correspondences and subsequently subtend higher accuracy matches. Figure 4.15 summarizes the RMSE values (in meters) for the Local Retrofit, Cross Correlation, Histogram, and SSIM methods. The results indicate that the Histogram technique consistently achieved the lowest RMSE across most datasets, followed closely by Cross Correlation and then SSIM. The Local Retrofit method recorded the highest RMSE values, especially in the DESERT dataset, since its requirement for crude detection and matching to maintain runtime meant it could not find enough good matches on the sparsest of datasets; It was excluded from further analysis.

Figure 4.16 compares the computational efficiency of each global matching technique across the five datasets. The Histogram technique demonstrated the fastest runtimes, followed closely by Cross Correlation. SSIM exhibited the longest runtimes of the remaining methods due to its more complex structural similarity calculations.

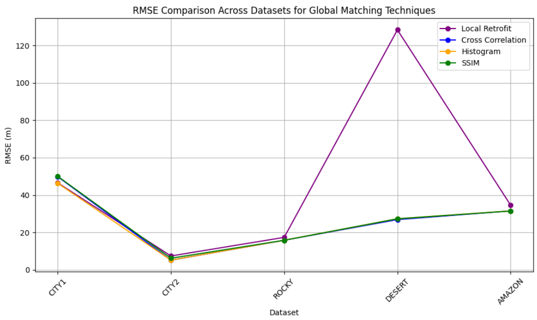


Figure 4.15: RMSE Comparison Across Datasets for Global Matching Techniques.

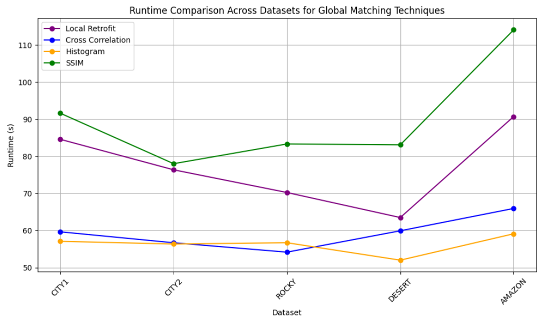


Figure 4.16: Runtime Comparison Across Datasets for Global Matching Techniques.

4.5.2. Considerations for GLobal Matching Techniques

This sections details the considerations when choosing the precision of the rotational alignment required for global matching. It subjects the estimated internal alignment to an additional skew. This was first tested under a 5-degree skew, where no changes occur. Figure 4.18 shows the percentage change in GPS error with a 10-degree skew. The results indicate that all methods are highly robust to small rotational offsets, with Cross Correlation demonstrating the highest robustness. This suggests that the choice of global matching technique can be prioritized for efficiency in the rotational estimation stage. Robustness testing evaluated each global matching technique's sensitivity to a

5- and 10-degree rotational offset. All matchers that made it this far saw no change in choice combination below a 5-degree offset. The resultant percentage deviation under a 10-degree offset is shown in Figure 4.18. Cross-correlation was the most robust, with the lowest percentage change in GPS error, from its prior estimate, across all datasets. Histogram and SSIM exhibited moderate robustness, while SSIM demonstrated the highest maximum error and variability. However, the ability of all matchers to maintain equivalent performance under offsets of up to 5 degrees suggests that they are all extremely robust to small rotational misalignments; This allows for prioritization of efficiency in the sections outputting the rotational estimate to this stage.

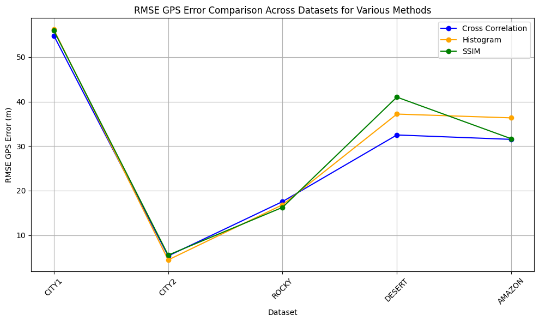


Figure 4.17: RMSE GPS Error Comparison Across Datasets for Various Methods

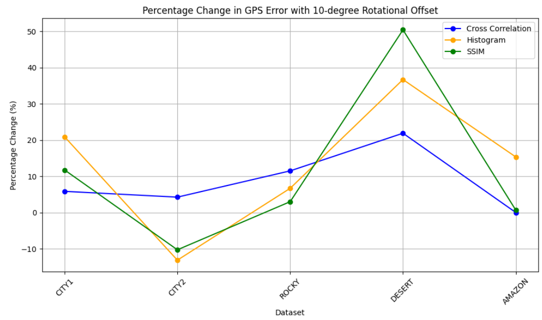


Figure 4.18: Percentage Change in GPS Error with 10-degree Rotational Offset

4.5.3. Final Selection of Global Matching Technique

Based on the comprehensive evaluation of accuracy, runtime, and robustness, the **Histogram** technique is identified and chosen as the most suitable global matching method for the system. Histogram consistently provided superior performance in terms of both RMSE and runtime, whilst maintaining sufficient robustness to rotational error.

4.6. Optimization Techniques

This section outlines the selected optimization methods employed to improve the performance of the UAV navigation system. The optimization techniques focus on filtering matches between images to balance noise and stability in the point sets.

4.6.1. Planar Transform Outlier Rejection Methods

Two outlier rejection methods, LMEDS (Least Median of Squares) and RANSAC (Random Sample Consensus), were evaluated for match filtration. Both methods performed near equivalently, with LMEDS displaying a slightly lower mean deviation. LMEDS was also significantly faster than RANSAC, making it the preferred choice for planar transform outlier rejection. The results are summarized in Figure 4.19 and Figure 4.20.

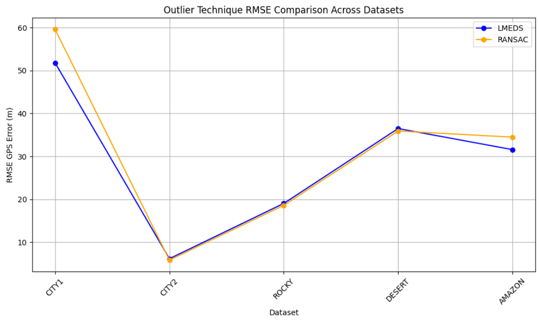


Figure 4.19: Radial GPS RMSE Comparison Across Datasets for LMEDS and RANSAC

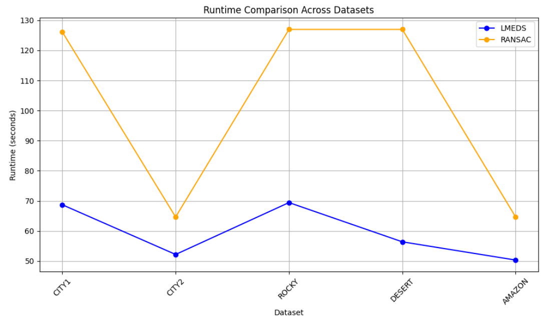


Figure 4.20: Runtime Comparison Across Datasets for LMEDS and RANSAC

4.6.2. Lowe's Ratio Test

Lowe's ratio test was employed to filter keypoint matches by comparing the distance of the best match to the second-best match. A static threshold proved inadequate for generalizing across diverse datasets, leading to inconsistent accuracy. To enhance robustness, a dynamic thresholding approach was implemented. The initial Lowe's ratio was set low and incrementally increased until a predetermined number of matches was achieved. The parameters chosen were 0.5 to 0.7 for the initial Lowe's ratio, 0.025 to 0.1 for the step size, 10 to 30 for the maximum iterations, and 2000 to 3000 for the minimum matches. These parameters were selected to balance generalization and match quality, ensuring a sufficient number of reliable matches without compromising accuracy. The dynamic Lowe's ratio filtering method was found to significantly enhance the quality of matches, improving the overall accuracy of the system.

4.6.3. N-Match or Absolute Thresholding

These approaches leveraged fixed number or descriptor distance thresholds to filter matches. However, without relativity to the density and quality of keypoints, empirical tests revealed that these methods added instability to the system. They either had no meaningful effect or removed too many matches, leading to loss of accuracy. The thresholds were deemed too sensitive for generalized applications.

4.7. Summary

The following methods were selected based on the comprehensive evaluation of accuracy, runtime, and robustness across diverse datasets:

- **Feature Detectors:** ORB for the crude layer and Dynamic AKAZE for the dense layer.
- **Local Feature Matcher:** FLANN for its superior runtime and scalability.
- **Transformational Estimator:** Rigid Transform SVD for its high accuracy, fast runtime, and robustness.
- **Image Similarity Estimator:** Histogram for its superior accuracy and runtime.
- **Optimization Techniques:** Dynamic Lowe's filtering for enhanced match quality and consistency. LMEDS was not included due to prior choices of methods not requiring it.

Chapter 5

Results

5.1. Detailed Performance Analysis

This section presents the results achieved by the navigation system under optimal conditions across various challenging datasets. The analysis focuses on the system's performance in terms of accuracy, efficiency, and generalizability, providing insights into its applicability in real-world UAV navigation scenarios.

xxx - show matches black and white. make a note linking setup for testing phase xxx

5.1.1. Key Performance Metrics

The system's generalized performance across five diverse datasets—CITY1, CITY2, ROCKY, DESERT, and AMAZON— is summarized with two key metrics: Localization radial error in RMSE m and percentage, and localization inference time in seconds across all datasets. The results are presented in Figure 5.1. Importantly, the system meets the real-time requirement of 2 seconds and the accuracy requirement of 10%.

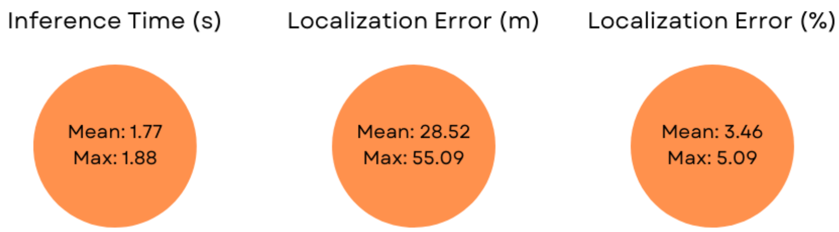


Figure 5.1: Key Accuracy And Runtime Performance Metrics

5.1.2. Flight Path Analysis

Figures 5.2 and 5.3 show the actual vs estimated flight of the UAV across the worst and best performing dataset, CITY1 and CITY2 respectively. Even in the worst performing dataset, the system maintains a relatively accurate flight path, relative to its movement size, with the estimated path closely following the actual path. This indicates the system's ability to generalize and gives an idea of the scope of the error in the system.

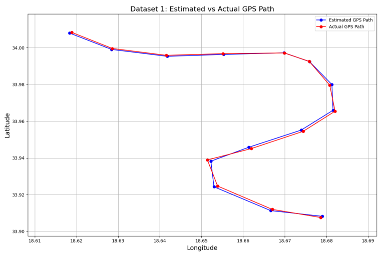


Figure 5.2: Flight Path of UAV in poorest performing Dataset

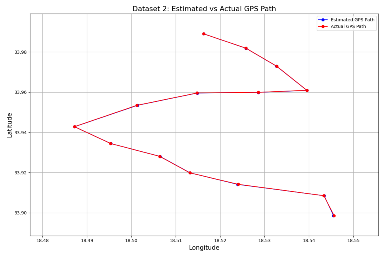


Figure 5.3: Flight Path of UAV in best performing Dataset

5.1.3. Detailed Performance Metrics

The detailed performance metrics include per dataset accuracy and time metrics. Namely, the radial RMSE error in GPS in meters and percentage, and the time taken for each stage of the pipeline, including the extraction and parameter inference time per while GPS is available, and the location inference time per image when GPS is lost. The results are summarized in Figures 5.4 and 5.4.

The results show that the system achieves consistent accuracy and runtime performance across each dataset. Further, its noted that both pipelines are able to maintain real-time performance, seen by the sum of the parameter inference and add times being below 2 seconds, as well as the location inference time being below 2 seconds. The accuracy of the system is also maintained, with the RMSE error being below 10% for all datasets.

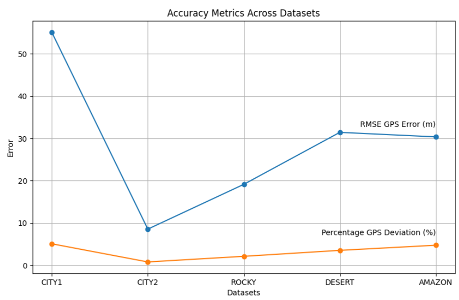


Figure 5.4: Accuracy Metrics for Each Dataset

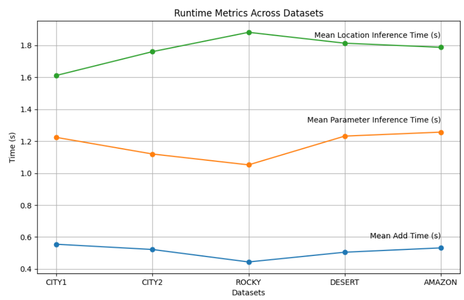


Figure 5.5: Runtime Metrics for Each Stage of the Pipeline

5.1.4. Dataset Performance Analysis

This section utilizes the violin plot, in Figure 5.6, to visualize the statistical distribution of radial errors across each dataset. This plots details the mean, median, and variance of the radial errors, as well as the distribution of errors across the datasets. The results provide insights into the system's accuracy and robustness in diverse environments.

The plot shows relatively consistent performance across datasets, with CITY2, being consistent with the global heading space, having the most accurate and reliable inter-dataset performance. The CITY1 and AMAZON datasets show comparitively high variation between their minimum and maximum errors, indicating a higher variance in direction.

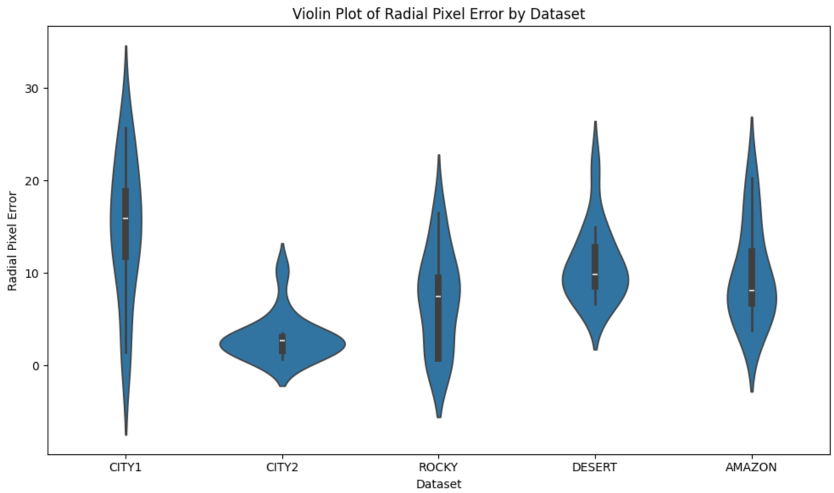


Figure 5.6: Plot of distribution of radial errors in each dataset

5.1.5. Error Distribution Map

The heatmap in Figure 5.7 visualizes the distribution of pixel deviations in the X and Y directions across the datasets. The results provide insights into the system's error patterns and highlight areas of potential improvement. The heatmap shows that the system has a relatively consistent error distribution across the datasets, with the majority of errors being within the 10-20 pixel range. However, some datasets, such as DESERT and AMAZON, exhibit larger errors of up to 30 pixels, indicating potential areas of significant error.

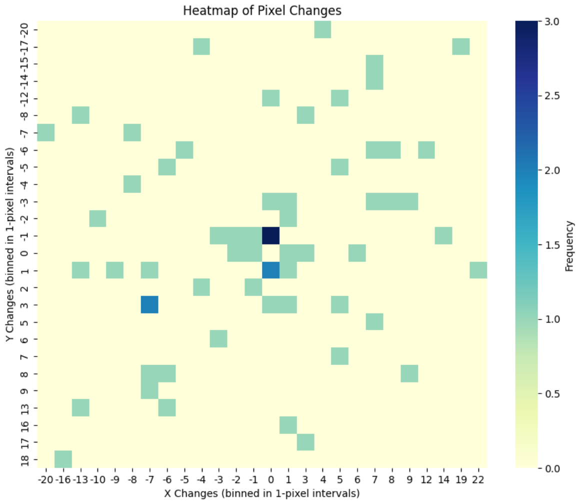


Figure 5.7: Heatmap of Pixel Deviations in X and Y Directions

5.1.6. An Account of Variance Between and Error in Datasets

Given the large variation in error across datasets, as seen in both the violin plot in figure 5.6 and the heatmap in figure 5.7, it is important to understand the sources of this variance. This section will detail the analysis conducted to identify the sources of this variance and the impact of these sources on the system's performance.

Internal Uncertainty and Variance

Correlations between error sizes and measures of uncertainty in the pipeline, specifically the standard deviation and mean-median difference in match vector magnitudes, were assessed. The variability in these vectors was used as a proxy for the uncertainty of the method. If the method was more uncertain when it was wrong, then the error likely originated from there. The match vectors, tested at various stages in the filtration process imply the accuracy of the transformational estimate. The results, in terms of correlation coefficients, as well as visual inspection, showed no relationship between uncertainty in the pipeline and error, indicating that there was not an increased likelihood of the variances in performance propagating from the anywhere including or before the transformational estimate.

Ground Truth Heading error and Variance

Upon observing a moderate correlation between error magnitude and image sequence in the dataset, the accuracy of the estimated headings from Google Earth data was reconsidered. Ground truth headings, as explained in 1.4, were estimated. These heading estimates were generated early in the project, prior to the development of an optimized rotational estimation technique. Additionally, this technique involved sequential estimation, introducing cumulative errors. To evaluate the impact of these inaccuracies, a newer, optimized pipeline for heading estimation was employed, which would significantly reduce, but not eliminate, ground truth heading errors on the AMAZON dataset; The dataset was retested. The results demonstrated a significant reduction in mean radial error from 30.37 m to 17.91 m. This improvement underscores that inaccurate ground truth heading data was a significant contributor to the system's variance in performance across datasets.

Impact of Heading Error on Accuracy

Given these findings, and the factor improvements, it is estimated that the true absolute maximum error for an individual image in the given datasets, with the ground truth data available, is in the range of 7.5-15 pixels radially; this corresponds to around 3.4%-6.8% of the magnitude of the translation vector for that image, remaining within acceptable limits for the system's design objectives. Similar impacts are expected on the accuracy results of the other datasets, excluding the CITY2 dataset which has a fixed North heading for all images. The absence of true ground truth heading data constituted the primary factor in the variance in accuracy across datasets.

Additional Sources of Error

While heading estimation errors were identified as the main contributor to the overall inaccuracies, several other factors also contribute to the errors observed:

Variations in Terrain and Environmental Conditions: Differences in terrain, lighting conditions, and feature density affect keypoint detection and matching accuracy. The system, even with dynamic parameter adjustments, cannot perfectly predict the expected quantity or quality of features without specific dataset tuning. Consequently, it is due to inherently miss some high-quality features or incorrectly match less reliable ones. Incorporating a machine learning model to predict expected feature density could enhance the system's performance.

Inaccuracies in Camera Parameter Estimations: While the system dynamically infers camera parameters, any inaccuracies in these estimations accumulate, leading to scaled errors during subsequent localization steps that rely on these parameters.

Image Noise and Resolution Limitations: The Google Earth images used are at a fixed resolution of 1920x1080 pixels, which inherently contain some noise. This noise can adversely affect feature detection and matching accuracy. **Algorithmic Constraints:** The selected image processing and matching algorithms have inherent limitations, particularly when handling extreme conditions or highly repetitive features. Additionally, being optimized for portability, they may not capture and represent every feature within an image perfectly.

Despite these factors, the achieved optimal radial error range, roughly 1-20 pixels across datasets, underscores the robustness and effectiveness of the image-based navigation system, affirming its suitability for practical UAV applications.

5.2. Stress Testing

This section involves the analysis of the system's robustness under various stressful conditions, including low-light environments and reduced mutual information. These tests were chosen practical and current relevance. The results aim to evaluate the system's performance under challenging scenarios and identify potential areas for improvement.

5.2.1. Mutual Information Results

In UAV navigation, accurate path maintenance is essential, especially when GPS signals are lost. Under these conditions, mutual information, or overlap—defined as the overlapping pixel information between current and reference images—plays a crucial role in guiding the UAV back on course. Specifically, when the UAV loses its path, there will be much less overlapping pixels than once the path is found.This study assesses whether reduced mutual information can still provide reliable navigational guidance and examines the trade-off between accuracy and computational efficiency.

The first objective was to identify the threshold of mutual information required to maintain an RMSE below 10%, ensuring navigational accuracy without inducing additional drift. Secondly, it was to evaluate if reducing mutual information by cropping image sizes can significantly decrease runtime while preserving acceptable accuracy levels.

Methodology

The CITY2 dataset is utilized, focusing exclusively on translational changes to simplify the mutual pixel calculation to the difference between the image size and translation vector. Images are progressively cropped from the default resolution of 1920x972 pixels to reduce mutual information. The analysis uses the mutual information as that of the image pair with the lowest mutual information, since it is the systems bottleneck. This ensures visual brevity, noting that mutual information of that of the image pair with the highest mutual information remains within 30% of the minimum across the dataset.

Results

Figure 5.8 shows that navigational accuracy remains stable until mutual information decreases to approximately 250,000 pixels (15.67% of a 1920x1080 image), below which accuracy sharply declines, indicating unreliable estimates. To maintain an RMSE below 10%, mutual information must exceed 300,000 pixels (18.81% of a 1920x1080 image).

Reducing mutual information by cropping images significantly decreases runtime. While the identified mutual information thresholds are specific to this dataset and pipeline, the results demonstrate the feasibility of balancing mutual information and runtime for effective navigation. Lastly, it is noted that the system achieved largely varying levels of accuracy at the same mutual information level, indicating different amounts of information is stored in different axes. It is therefore recommended, if this method of decreasing runtime is considered, that the image is cropped towards an aspect ratio of one to maintain stability as the UAV rotates.

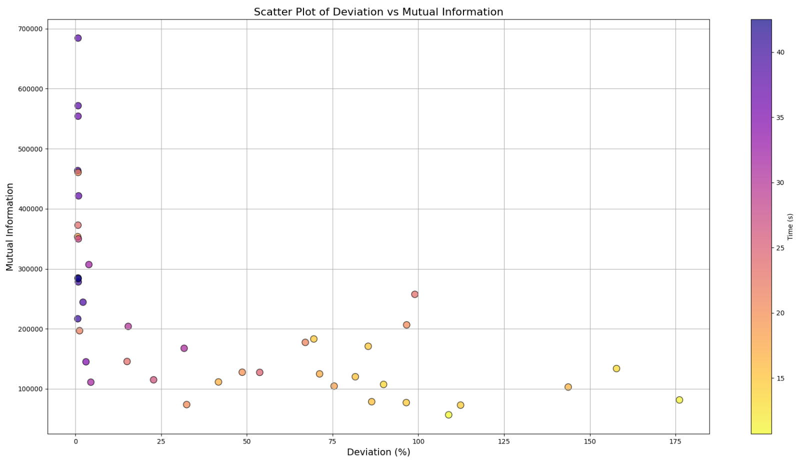


Figure 5.8: Accuracy vs Mutual Information

Conclusions

This study demonstrates that UAV navigational accuracy is maintained with mutual information levels up to approximately 20% of the original image size. Above this threshold, additional mutual information yields diminishing returns in accuracy, indicating an inelastic performance region. Runtime performance improves proportionally with reduced mutual information, presenting a trade-off between accuracy and computational efficiency. For instance, cropping the image resolution to 1100x635, subtending 300,000 mutual pixels achieves a 1.8-fold runtime reduction with only a 0.1% increase in error. These findings emphasize the importance of calibrating resolution levels to balance navigational accuracy and real-time performance. Further, when applying this method, cropping the image towards an aspect ratio of one is recommended to maintain stability as the UAV rotates. Future work should explore adaptive methods that start with reduced mutual information and increase it only when necessary, such as when the UAV is turning around to find its path.

5.2.2. Low-Light Testing

UAV navigation performance can be significantly impacted by low-light conditions, where keypoint detection and matching accuracy decline due to reduced visibility and increased noise. This study evaluates the robustness of the navigation method under simulated evening and nighttime conditions across five datasets: CITY1, CITY2, ROCKY, DESERT, and AMAZON. Conditions are simulated through adaptions in image brightness, contrast, tint, and noise levels to mimic real-world low-light environments. Evening conditions involve moderate applications of these adjustments, while nighttime conditions feature severe reductions in visibility and increased noise levels.

The objectives of this study are twofold: firstly, to assess the robustness of the UAV navigation system under low-light conditions by evaluating how evening and nighttime lighting affect the accuracy of navigational estimates; and secondly, to compare the system's performance across diverse environments with varying feature densities under these low-light scenarios.

Representative examples from the CITY1 dataset illustrate these conditions:

Figure 5.9: Daytime Image of Cape Town

Figure 5.10: Evening Image of Cape Town

Figure 5.11: Night Image of Cape Town

Figure 5.12: Lighting Conditions in Cape Town

Results

Figure 5.13 summarizes the accuracy of the navigation method under evening and nighttime conditions for each dataset. Nighttime performance shows that the CITY and ROCKY datasets maintain relatively low RMSE values, indicating resilience to severe low-light environments, while the DESERT and AMAZON datasets exhibit substantial increases in RMSE due to sparse and repetitive features hindering effective keypoint matching. Evening conditions result in minimal degradation of navigational accuracy for most datasets, with the DESERT dataset experiencing a moderate error increase from its initial RMSE of 31.44 m to 156.34 m, corresponding to a 17.51% RMSE error, which remains usable.

The performance disparity between datasets underscores the importance of feature density in low-light navigation, with environments rich in distinct landmarks (e.g., CITY and ROCKY) demonstrating greater robustness compared to feature-poor environments (e.g., DESERT and AMAZON).

Overall, the navigation method demonstrates profound resilience in well-featured environments under low-light conditions. However, challenges still remain in feature-sparse environments during nighttime. The usage of additional sensors, such as infrared cameras, could enhance navigational accuracy in these challenging scenarios.

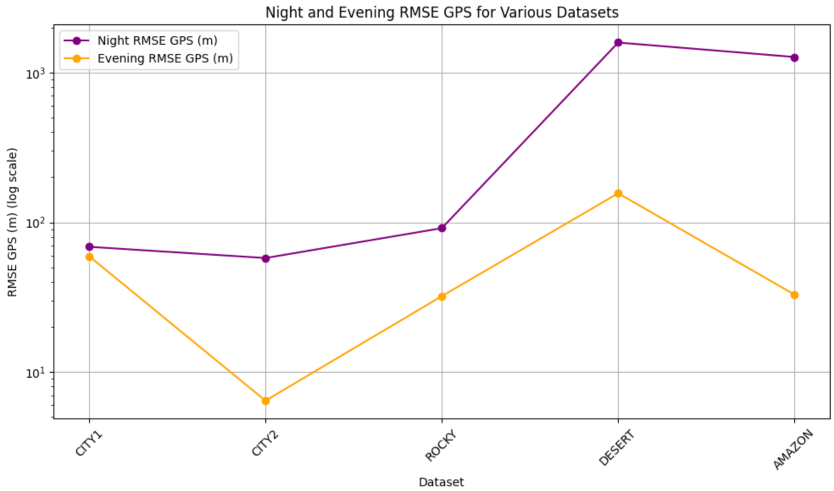


Figure 5.13: Night and Evening RMSE GPS for Various Datasets

5.3. Results Conclusion

The results presented in this chapter confirm that the system meets the accuracy and time requirements established in the objectives. The system demonstrated reliable performance across various datasets, including sparse-feature environments, showcasing its ability to generalize effectively. Despite practical limitations such as resolution constraints, performance limitations, and fluctuating terrain feature densities, the system maintained robust accuracy.

The analysis further indicates that accuracy would be significantly improved with access to accurate ground truth headings and known camera parameters, reducing potential sources of error. Furthermore, the system showed resilience under challenging conditions, including low-light scenarios and images with reduced mutual information. Even with limited pixel overlap, the system continued to deliver reliable navigation data.

Overall, the system has proven to be a versatile and effective solution for UAV navigation, capable of sustaining performance under a wide range of operational challenges and environmental conditions.

Chapter 6

Summary and Conclusion

Global Positioning System (GPS) vulnerabilities, such as jamming and spoofing, pose significant risks to Unmanned Aerial Vehicle (UAV) navigation, potentially leading to loss of control and mission failure. Recognizing the limitations of existing alternatives—which are often prohibitively expensive or unreliable—this project addressed the critical need for a robust, GPS-independent navigation solution for UAVs operating in GPS-denied environments.

The primary objective was to develop an accurate and generalizable image-based GPS localization system. To achieve this, the project optimized the localization pipeline by selecting and integrating effective feature extraction, matching, and planar transformation estimation techniques. The system was designed to estimate GPS locations using only prior telemetry data and image features, ensuring radial errors remained below 10% of the UAV's displacement from a reference image. Additionally, the system was engineered for real-time operation, achieving response times under two seconds following GPS signal loss, thereby allowing pilots to maintain effective manual control without significant delays. Comprehensive validation across diverse environments confirmed the system's generalizability and robustness, eliminating the need for environment-specific parameter tuning.

The implementation successfully met all outlined objectives. The image-based localization system demonstrated high accuracy and real-time performance across multiple datasets, including challenging sparse-feature environments. It maintained robust accuracy despite practical limitations such as resolution constraints, varying terrain feature densities, and fluctuating environmental conditions. The system's resilience was further evidenced under low-light conditions and scenarios with limited pixel overlap, ensuring reliable navigation data delivery even in adverse situations.

The successful development and validation of this system signify a substantial advancement in UAV navigation technology. By providing a reliable alternative to GPS, the system enhances UAV operational safety and effectiveness in both military and civilian applications.

However, the project also identified several areas that warrant further research and development. The system was unable to consistently balance the quantity (stability) and quality (outliers) of feature detection across diverse feature landscapes, highlighting the need for more sophisticated algorithms or machine-learning techniques. Additionally, sparse-feature environments in low-light conditions continue to pose significant challenges. Furthermore, this study did not account for distortions resulting from scale and perspective changes, necessitating the application and testing of homography transformations to address these distortions effectively.

In summary, this project successfully developed a versatile and effective image-based GPS localization system for UAVs, addressing the critical need for reliable navigation in GPS-denied environments. The system has proven to be practical for real-world applications, demonstrating robust performance across a wide range of operational conditions. With further enhancements to improve accuracy and overcome existing limitations, this localization system holds substantial promise for enhancing UAV navigation reliability in both military and civilian sectors, supporting a broader scope of missions with increased safety.

Chapter 7

Future Work

This study establishes a foundational framework for image-based UAV navigation. While effective, several advancements could enhance the accuracy, practicality, and efficiency of the system in future implementations.

7.0.1. Multi-Image Weighted Inference System

Future implementations could improve location accuracy by using a weighted estimation approach based on multiple reference images, rather than relying on a single image. Aggregating data from multiple images would provide a more reliable position estimate, mitigating the effects of individual image distortions or outliers. Although this method was not tested in this study due to time constraints, it has the potential to enhance the robustness of location estimation with minimal increase in computational load.

7.0.2. Non-Planar Stereo Matching

For navigating complex terrains with non-planar ground surfaces, incorporating a non-planar stereo matching approach could improve depth perception and accuracy. By accounting for variations in surface elevation, the UAV could more accurately approximate altitude, enhancing its location estimates over diverse landscapes.

7.0.3. Integration with Mapping and Reference Images

Integrating reference images with up-to-date mapping data could eliminate the need for prior image capture and a fixed return path to base. This approach would enable the UAV to navigate freely within a mapped region without accumulating positional drift. Using prior flight data or collaboratively obtained maps would allow the system to maintain accuracy over a larger operational area, providing extreme reliability in the event of GPS signal loss.

7.0.4. Distortion Corrections

Although this system did not account for the effects of roll and pitch changes or altitude variations, these distortions are common in real-world applications. Future implementations could integrate correction mechanisms for these factors to enhance accuracy. Solutions such as OpenCV's homography estimators, offer built-in methods for compensating for such distortions, allowing easy to implement, efficient real-time correction on compatible devices.

7.0.5. Implementation on a Single Board Computer (SBC)

To improve system performance and enable more complex computations, future work could explore implementing the system on a higher-performance Single Board Computer (SBC) with enhanced processing power. A more capable SBC would support the integration of multiple estimation techniques, facilitating more sophisticated data fusion to enhance overall accuracy.

7.0.6. Higher Resolution Imaging

Incorporating higher-resolution imaging in future systems could significantly improve feature extraction and matching accuracy by capturing more detailed visual information. However, as resolution increases, so do the processing and memory demands. To maintain real-time performance, careful management of computational load and memory usage will be essential.

In summary, the proposed future work addresses a range of enhancements, from improving altitude and position inference to optimizing system performance and accuracy through advanced image processing and storage management techniques. These improvements would ensure a more robust, adaptable, and scalable navigation system suitable for various UAV applications.

XXX - testing real world data. Also testing it against Google Earth Data to see how time affects estimation at varying times.

Bibliography

[1] Google LLC, “Google earth,” https://earth.google.com/, 2024, accessed: 2024-10-29.

[2] J.-W. Bian, W.-Y. Lin, Y. Liu, L. Zhang, S.-K. Yeung, M.-M. Cheng, and I. Reid, “Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence,” *International Journal of Computer Vision*, vol. 128, June 2020.

[3] M. Weiss, “Personal communication,” RF Engineer, 2024, june 26.

[4] Geotab Team, “What is gps?” 2024, accessed: October 17, 2024. [Online]. Available: https://www.geotab.com/blog/what-is-gps/

[5] Y. Zhuang, X. Sun, Y. Li, J. Huai, L. Hua, X. Yang, X. Cao, P. Zhang, Y. Cao, L. Qi, J. Yang, N. El-Bendary, N. El-Sheimy, J. Thompson, and R. Chen, “Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches,” *Information Fusion*, vol. 95, pp. 62–90, 2023.

[6] Scout Aerial Australia, “An introduction to lidar technology and its applications,” 2024, accessed: October 17, 2024. [Online]. Available: https://www.scoutaerial.com.au/article-lidar/

[7] J. Brewer, “Line of sight requirements for reliable low-power rf communication,” White Paper, ATEK Access Technologies, 2024, senior Electrical Design Engineer.

[8] M. J. Wright, L. Anastassiou, C. Mishra, J. M. Davies, A. M. Phillips, S. Maskell, and J. F. Ralph, “Cold atom inertial sensors for navigation applications,” *Frontiers in Physics*, vol. 10, p. 994459, 2022.

[9] P. O’Donovan, “Optical flow: Techniques and applications,” 2005, accessed: 2024-10-29. [Online]. Available: http://dgp.toronto.edu/~donovan/stabilization/opticalflow.pdf

[10] P. Barnum, B. Hu, and C. Brown, “Exploring the practical limits of optical flow,” no. Technical Report 806, August 2003.

[11] D.-G. Sim, R.-H. Park, R.-C. Kim, S. U. Lee, and I.-C. Kim, “Integrated position estimation using aerial image sequences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 1–18, Jan 2002. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=982881

[12] X. Zhang, F. Xiao, M. Zheng, and Z. Xie, "Uav image matching from handcrafted to deep local features," *European Journal of Remote Sensing*, vol. 57, no. 1, 2024. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e074c475d93d5e79ded55b5554ea57cb5fb71207

[13] M. Trajkovic and M. Hedley, "Fast corner detection," *Image and Vision Computing*, vol. 16, pp. 75–87, Feb 1998, accessed: 2024-10-29. [Online]. Available: https://www.researchgate.net/publication/223831601_Fast_Corner_Detection/citation/download

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," vol. 6314, pp. 778–792, 2010, accessed: 2024-10-29. [Online]. Available: https://researchgate.net/publication/221304115_BRIEF_Binary_Robust_Independent_Elementary_Features/citation/download

[15] S. A. K. Tareen and Z. Saleem, "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2018, pp. 1–10, accessed: 2024-10-29. [Online]. Available: https://ieeexplore.ieee.org/document/8346440

[16] G. Reddy, C. Vani, and E. Krishna, "Implementation of mldb as high performance machine learning database for high performance applications," *Materials Today: Proceedings*, Mar 2021, accessed: 2024-10-29. [Online]. Available: https://www.researchgate.net/publication/349871153_Implementation_of_MLDB_as_high_performance_machine_learning_database_for_high_performance_applications/citation/download

[17] OpenCV, *cv::AKAZE Class Reference*, 2018. [Online]. Available: https://docs.opencv.org/3.4/d8/d30/classcv_1_1AKAZE.html

[18] Rpaultrat, "Efficient neural feature detector and descriptor," GitHub, 2023, accessed: 2024-10-24. [Online]. Available: https://github.com/rpaultrat/SuperPoint

[19] Cvg, "Lightglue: Local feature matching at light speed," GitHub, 2023, accessed: 2024-10-24. [Online]. Available: https://github.com/cvg/LightGlue

[20] OpenCV, *cv::BFMatcher Class Reference*, 2022. [Online]. Available: https://docs.opencv.org/4.x/d3/da1/classcv_1_1BFMatcher.html

[21] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. X, 2014, accessed: 2024-10-29. [Online]. Available: https://www.cs.ubc.ca/research/flann/uploads/FLANN/flann_pami2014.pdf

[22] OpenCV, *Feature Matching in OpenCV*, 2018. [Online]. Available: https://docs.opencv.org/3.4/dc/dc3/tutorial_py_matcher.html

[23] Z. Rezvani, S. Shekarizeh, and M. Sabokrou, "Global-local processing in convolutional neural networks," 2023. [Online]. Available: https://doi.org/10.48550/arxiv.2306.08336

[24] V. Sharma, "Image-template matching using cross-correlation," *Medium*, 2022, accessed: 2024-10-26. [Online]. Available: https://vipin-sharma.medium.com/image-template-matching-using-cross-correlation-2f2b8e59f254

[25] A. Rosebrock, "How-to: 3 ways to compare histograms using opencv and python," *PyImageSearch*, 2014, accessed: 2024-10-26. [Online]. Available: https://pyimagesearch.com/2014/07/14/3-ways-compare-histograms-using-opencv-python/

[26] ——, "Image difference with opencv and python," *PyImageSearch*, 2017, accessed: 2024-10-26. [Online]. Available: https://pyimagesearch.com/2017/06/19/image-difference-with-opencv-and-python/

[27] OpenCV, "Warp affine transformation," 2024, accessed: 2024-10-24. [Online]. Available: https://docs.opencv.org/4.x/d4/d61/tutorial_warp_affine.html

[28] O. Sorkine-Hornung and M. Rabinovich, "Least-squares rigid motion using svd," 2017, january 16, 2017.

[29] OpenCV, "Homography estimation," 2024, accessed: 2024-10-24. [Online]. Available: https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html

[30] R. B. Fisher, "The ransac (random sample consensus) algorithm," 2002, accessed: 2024-10-28. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FISHER/RANSAC/

[31] D. S. Farin, "Automatic video segmentation employing object/camera modeling," Ph.D. dissertation, Technische Universiteit Eindhoven, 2005.

Appendix A

Project Planning Schedule

This is an appendix.

Appendix B

Outcomes Compliance

Student's Graduate Attribute (GA) Achievement Plan

GA 1: Problem Solving

The navigation system I developed effectively addresses complex engineering and software problems to achieve a robust and accurate image-based localization system. By leveraging mathematical and statistical methods, I crafted a system that accurately identifies, matches, and estimates the transformations of the UAV even in challenging GPS-denied environments. Throughout the project, I prioritized methods that could reliably extract features and estimate transformations in real time, allowing the UAV to autonomously navigate with precision. I rigorously tested, through over 10,000 lines of code, various algorithms and parameters to ensure high accuracy, outlier rejection, and robustness across different environmental conditions. Furthermore, throughout the project, I faced numerous hurdles regarding issues in the accuracy and runtime of the solution; I had to perform extensive debugging, address prior assumptions, and explore different ways to solve problems. For instance, normalizing images to the global space might seem straightforward, but after noting errors, I realized I had to find their source. Rotational loss was found to be minimized when alignment was applied between images only. Additionally, I used literature to identify innovative approaches that could be adapted to overcome specific challenges related to feature scarcity, low-light conditions, and image warping.

GA 2: Application of Scientific and Engineering Knowledge

My project extensively utilized computer vision theories, image processing techniques, and pre-trained machine learning models to develop an effective navigation system. I applied various algorithms grounded in computer vision to accurately detect and analyze landscape changes. Leveraging principles of homographic transformations, I was able to account for the UAV's motion and orientation without relying on external GPS signals. Furthermore, I incorporated statistical methods to quantify model output and to improve accuracy by systematically testing different outlier rejection methods. This application of scientific and engineering knowledge was instrumental in developing a system that could operate reliably across multiple terrains, validating the project's approach to image-based navigation.

GA 3: Engineering Design

The engineering design process began with creating a high-level flow diagram that outlined each stage in the image processing pipeline. This design incorporated multiple resources and techniques and integrated them effectively with robust design measures to handle inaccuracies. I utilized my knowledge of engineering principles gained during the course of my studies in design skills and software development to design a modular, robust system capable of adapting to different environments. This structured approach allowed the system to maintain stability and precision, even in scenarios with sparse image features or environmental variations, ultimately fulfilling the project's goals of reliable UAV navigation.

GA 4: Investigations, Experiments, and Data Analysis

To support the development and validation of the system, I collected extensive data across various environments and under different conditions. I systematically tested the navigation system using diverse datasets and conditions, such as sparse desert landscapes and low-light repetitive terrain, to identify the suitability, generalizability, accuracy, and robustness of different techniques. Using statistical analysis, I was able to quantify the variance in results, leading to the subsequent discovery of faults in logic and techniques, as well as understanding the true performance of the system. The ability to leverage systematic investigation and experimentation, as well as my knowledge of data analytics, allowed me to implement, test, and optimize various methods with limited guidance. The project required ongoing experimentation with different feature detection and matching algorithms, including neural network-based methods, to fine-tune the system's performance. The large array of parameters needed to be tuned meant I could not simply test every possible solution; I needed to systematically investigate from principles and then experiment based on those to ensure efficiency and not forgo a better alternative.

GA 5: Engineering Methods, Skills, and Tools (including IT)

I employed a wide range of tools and engineering methods, including Python libraries such as OpenCV and advanced frameworks available on GitHub for image processing. Furthermore, this project involved rigorous research into the best and most current solutions for all stages of the pipeline. By using the latest methods, I was able to demonstrate the applicability of this system in difficult conditions, something made possible by recent advancements in computer vision techniques. This project also required rigorous adherence to software engineering practices, ensuring modularity, code optimization, and thorough error handling. I adopted computational techniques to evaluate the system's accuracy and performance, allowing for the assessment and validation of methods in a controlled environment. By following best practices and leveraging tools to address issues in real time, I was able to identify areas for improvement and continuously optimize the system's accuracy and response.

GA 6: Professional and Technical Communication

Throughout the project, I demonstrated strong communication skills by preparing both written reports and oral presentations. These presentations communicated complex technical concepts in a clear and concise manner to stakeholders and supervisors. Additionally, I documented my methods and findings in detail, ensuring reproducibility and clarity for future work. The project's outcomes were communicated effectively in both technical and non-technical formats, highlighting my ability to translate technical progress and results into comprehensible information.

GA 8: Individual Work

As the lead on this project, I took primary responsibility for all aspects of the development and implementation process. This included problem-solving, design, and optimization, as well as the testing and analysis of results. My role required me to work independently to research, develop, and refine solutions, utilizing various resources and overcoming challenges without external assistance. This project reflects my commitment to delivering high-quality work and achieving the outlined objectives independently.

GA 9: Independent Learning Ability

This project demanded a high level of independent learning to solve complex navigation challenges in GPS-denied environments. I actively sought out research materials, consulted technical documentation, and applied knowledge from relevant scientific literature. I expanded my understanding of image processing techniques, statistical analysis, and feature extraction methods independently, equipping myself with the skills necessary to achieve project goals without requiring extensive guidance. This process underscored my ability to learn independently and apply new knowledge effectively to overcome project-specific challenges.