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.

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

Declaration i

Abstract ii

List of Figures vi

List of Tables viii

Nomenclature ix

1. Introduction 1

1.1. Background . . . 1
1.2. Problem Statement . . . 3
1.3. Aims and Objectives . . . 3
1.3.1. Aim . . . 3
1.3.2. Objectives . . . 3
1.4. Scope . . . 5
1.5. Data Provisioning . . . 6
1.6. Summary of Work . . . 6
1.7. Structure of the Report . . . 7

2. Literature Review 8

2.1. Related Work . . . 8
2.2. Background . . . 9
2.2.1. Feature Detectors . . . 9
2.2.2. Feature Matching . . . 11
2.2.3. Image Similarity Computation . . . 12
2.2.4. Planar Transformation Estimators . . . 13
2.2.5. Optimization Techniques . . . 15

3. System Design 17

3.1. System Pipeline . . . 17
3.1.1. Pipeline Stages . . . 17
3.2. Dynamic Methods and Techniques . . . 20
3.3. Testing Shortlist . . . 21

3.3.1. Software . . . 21
3.4. Conclusion . . . 22

4. Comparison of Methods **23**
4.1. Testing Setup . . . 23
4.1.1. Critical Testing Pipeline Understanding . . . 23
4.1.2. Datasets . . . 23
4.1.3. Testing Structure . . . 25
4.2. Feature Detectors . . . 26
4.2.1. Accuracy and Runtime . . . 27
4.2.2. Final Selection of Feature Detectors . . . 27
4.3. Local Feature Matchers . . . 28
4.3.1. Accuracy and Runtime Evaluation . . . 28
4.3.2. Robustness Testing . . . 29
4.3.3. Considerations . . . 29
4.3.4. Final Selection of Local Feature Matcher . . . 30
4.4. Planar Transform Estimators . . . 30
4.4.1. Accuracy and Runtime Evaluation . . . 30
4.4.2. Robustness Testing . . . 31
4.4.3. Final Selection of Rotational Estimator . . . 31
4.5. Image Similarity Estimators . . . 31
4.5.1. Accuracy and Runtime Evaluation . . . 32
4.5.2. Robustness Testing . . . 32
4.5.3. Considerations for Alignment Prior to GLobal Matching . . . 33
4.5.4. Final Selection of Global Matching Technique . . . 33
4.6. Optimization Techniques . . . 34
4.6.1. Planar Transform Outlier Rejection Methods . . . 34
4.6.2. Lowe's Ratio Test . . . 34
4.6.3. N-Match or Absolute Thresholding . . . 35
4.7. Summary . . . 36

5. Results **37**
5.1. Key Metric Analysis . . . 37
5.1.1. Key Performance Metrics . . . 37
5.1.2. Flight Path Analysis . . . 39
5.1.3. Error Distribution Map . . . 40
5.2. Stress Testing . . . 42
5.2.1. Low Resolution Testing . . . 42
5.2.2. Dynamic Lighting Testing . . . 45
5.2.3. Mutual Information Results . . . 45

5.2.4. Low-Light Testing 47
5.3. Results Conclusion 49

6. Summary and Conclusion **51**

7. Future Work **53**
7.0.1. Multi-Image Weighted Inference System 53
7.0.2. Non-Planar Stereo Matching 53
7.0.3. Integration with Mapping and Reference Images 53
7.0.4. Distortion Corrections 54
7.0.5. Implementation on a Single Board Computer (SBC) 54
7.0.6. Higher Resolution Imaging 54

Bibliography **55**

A. Project Planning Schedule **58**

B. Outcomes Compliance **59**

List of Figures

2.1. Feature Detection and Matching; adapted from [1]. 10
2.2. Reproduced from [2]. 16

3.1. High-Level Flow of the System 17

4.1. Examples of the CITY1 and CITY2 Datasets. 25
4.2. Example of the ROCKY Dataset. 25
4.3. Example of the DESERT Dataset. 25
4.4. Example of the AMAZON Dataset. 25
4.5. Overview of Datasets 25
4.6. Radial Error for Various Local Detectors. 27
4.7. Runtime for Various Local Detectors. 27
4.8. RMSE and Runtime Comparison for ORB, AKAZE, and SuperPoint Across Datasets. 27
4.9. Radial Error for BFMatcher and FLANN. 28
4.10. Runtime Comparison for BFMatcher and FLANN. 28
4.11. Convergence in RMSE Lat-Lon Error Between FLANN and BFMatcher Across Keypoint Targets. 29
4.12. RMSE Comparison Across Datasets for Planar Transform Estimators. . . . 31
4.13. Runtime Comparison Across Datasets for Planar Transform Estimators. . . 31
4.14. RMSE and Runtime Comparison Across Datasets for Planar Transform Estimators. 31
4.15. RMSE Comparison Across Datasets for Global Matching Techniques. . . . 32
4.16. Runtime Comparison Across Datasets for Global Matching Techniques. . . 32
4.17. Percentage Change in Lat-Lon Error with 10-degree Rotational Offset . . . 33
4.18. Radial Lat-Lon RMSE Comparison Across Datasets for LMEDS and RANSAC. 34
4.19. Runtime Comparison Across Datasets for LMEDS and RANSAC. 34

5.1. Flight Path of UAV in Rocky Dataset (Worst) 40
5.2. Flight Path of UAV in Desert Dataset (Best). 40
5.3. Heatmap of Pixel Deviations in X and Y Directions 40
5.4. Mutual Information vs Mean Error Percentage 46
5.5. Mutual Information vs Mean Localization Time 46
5.6. Daytime Image of Cape Town 48

5.7. Early Evening Image of Cape Town 48
5.8. Late Evening Image of Cape Town 48
5.9. Night Image of Cape Town 48
5.10. Lighting Conditions in Cape Town 48
5.11. Mean Radian Increase in Error Under Nighttime and Evening Conditions. 49

List of Tables

5.1. **Mean Radial Errors** 39
5.2. **Maximum Radial Errors** 39
5.3. **Mean Processing Times (Seconds)** 39
5.4. **Maximum Processing Times (Seconds)** 39
5.5. Resolution vs Mean Radial Percent and Mean Location Inference Time . . 43
5.6. DATSETROT2: Resolution vs Mean Radial Percent and Mean Location Inference Time 43
5.7. DATSETCPT2: Resolution vs Mean Radial Percent and Mean Location Inference Time 43
5.8. DATSETROCK2: Resolution vs Mean Radial Percent and Mean Location Inference Time 44
5.9. DATSETSAND2: Resolution vs Mean Radial Percent and Mean Location Inference Time 44
5.10. DATSETAMAZ2: Resolution vs Mean Radial Percent and Mean Location Inference Time 44
5.11. Parameter Settings for Different Lighting Effects 47
5.12. Mean Radial Percentages for Different Datasets under Varying Lighting Conditions 49

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.
Lat-Lon Estimation	Estimation of Latitude and Longitude; If Referencing an error, it is the radial one.

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	Global Navigation Satellite System
DOF	Degrees of Freedom

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 [5]. 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 [5].

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 [6]. 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 [7]. 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 [8]. 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 [9].

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 associated telemetry data, either through an acquired database or by capturing images during flight prior to GNSS signal loss, to estimate their position and heading. This type of 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 [10].

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: [9]. LIDAR [8]. RF [7] Odometry [6]. 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 [4]. Current alternatives to GNSS navigation, such as quantum sensing, odometry, RF communication, and LIDAR, are either too expensive, bulky, reduce stealth capabilities, or suffer from drift over time [6–9]. To enhance the safety and effectiveness of national military missions, a comprehensive evaluation of a novel, image-based, redundancy navigation, system is required. Existing studies on image-based UAV navigation lack detailed system setups and fail to test performance across diverse environments and challenging conditions [11, 12].

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 [11].. 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 [13].

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.

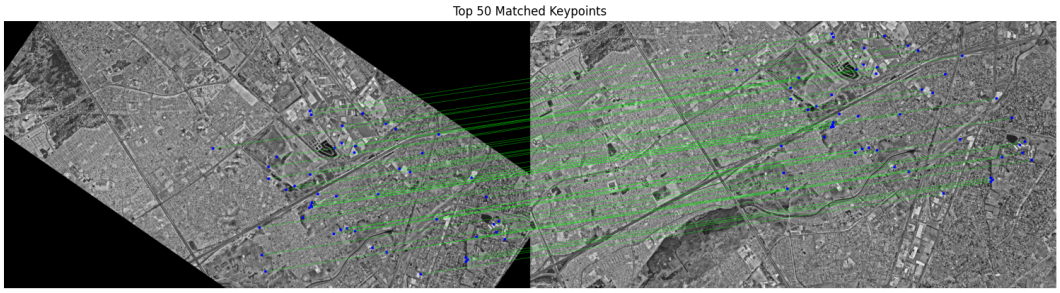


Figure 2.1: Feature Detection and Matching; adapted from [1].

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** [14] keypoint detector with the **BRIEF** [15] 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 [16].

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)** [17] 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 [18]. However, a notable strength of AKAZE is its its rotational invariance [16].

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 [19]. 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 [20]. 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 [19].

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 [21].

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 [22].

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 [20]. 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 [23].

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 [23].

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 [23].

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 [24]. 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 [25].

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 [26].

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 [27].

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 [24]. 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 [28]. 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 [29].

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 [28].

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 [30]. 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 [31]. 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 [31].

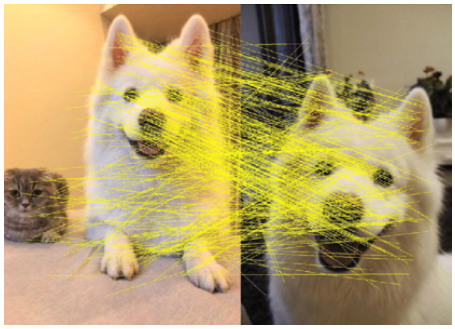
Local Maxima Extrema Density Selection (LMEDS) for Planar Transformation

LMEDS is a keypoint selection method designed to prioritize areas of high feature density by identifying local maxima as keypoints for matching [32]. 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.

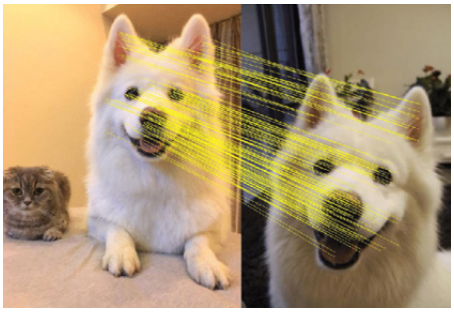
Lowe's Ratio Test

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 after GNSS signal is lost. The high-level flow of the system is illustrated in Figure 3.1.

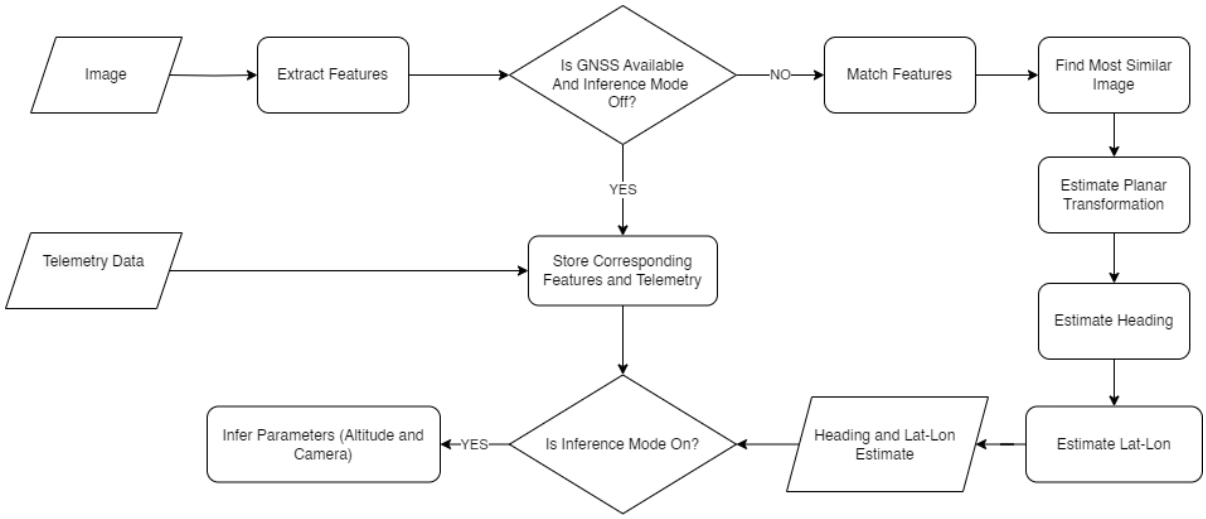


Figure 3.1: High-Level Flow of the System

3.1. System Pipeline

The system pipeline consists of multiple stages, each designed to perform a specific task in the navigation process. The pipeline is designed to be robust, accurate, and computationally efficient, ensuring reliable navigation in various operational scenarios. The pipeline stages are detailed below.

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 GNSS 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

3.3.1. Software

This sections gives a brief idea of the flow of software used in the system. This snippet will be found in the main loop and details the key flows. The full code may be found at https://github.com/Samshabz/Skripsie

```
# Setup Lines...

# Phase 1: GNSS is Available and Inference Mode is On
# Add images from the dataset and infer Pixel to Metre Factor
for i in range(1, inference_images + 1):
    navigator.add_image(i, directory)
navigator.estim_pos(inference_images, Inference_Mode_On=True)
navigator.find_pixel_to_metre_factor()

# Phase 2: GNSS is Available and Inference Mode is Off
# Add the rest of the images from dataset
for i in range(inference_images+1, total_images + 1):
    navigator.add_image(i, directory)

# Phase 3: GNSS is Unavailable
# Estimate the position of the UAV
navigator.estim_pos(total_images, Inference_Mode_On=False)

# Debug Lines...
```

3.4. Conclusion

The system design integrates feature extraction, matching, and transformation estimation to enable precise UAV navigation in the absence of GNSS signal. 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 explanations of the testing setup are important to understand the subsequent, results section.

4.1.1. Critical Testing Pipeline Understanding

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

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, subtke perspective and scale distortions were present due to 3-Dimensional rendering in Google Earth. The magnitude of these subtle distortions is implicitly inferred by the degrees of freedom, the number of distortion types a method accounts for, of that of the best performing transformational estimation technique, negating the strict requirement for quantification thereof.

To ensure maximal usage of the dataset, the same images are used in both the with GNSS signal and no GNSS signal stages. In the GNSS Available stages, all 15 images are streamed and added, storing their features and telemetry. The first 5 images are used to infer the fixed pixel-to-meter factor, related to the camera focal length and UAV altitude. Thereafter, in the no GNSS signal stages, the 15 images have their location and heading estimated. Importantly, each image in the no GNSS signal stages recomputes its features and does not infer its location based on its own image, which would have no displacement, from the with GNSS signal stages, ensuring a fair test. Additionally, the images are spaced sufficiently to ensure that enough challenge is present in the datasets.

The datasets have a variety of properties to ensure that the tests are challenging but not impossible and, more broadly, to ensure they mimic real-world data. The datasets are captured at ground heights of 5–7 km, with a resolution of 1920×972 pixels; the latter choice was required to eliminate image text that would cause false positive matches. The radial movement between frames varies between 300 and 700 pixels, depending on the image and the best match found. This corresponds to around 65 to 90% overlap. Even though this means the measurement of metres is dependent on this translation size, it is true that the method subtending the lowest metre error also subtends the lowest overall error. In other words, this variability does not impact inter-method comparisons using metres. Each dataset consists of 15 images, balancing testing time and ensuring sufficient evaluation of 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.

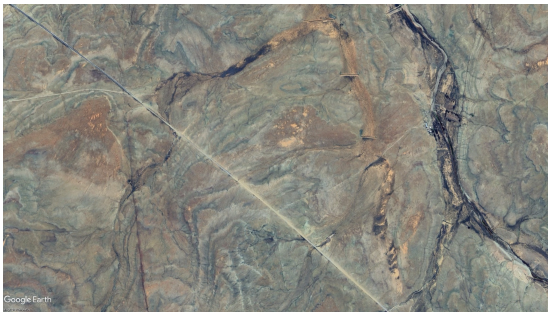


Figure 4.2: Example of the ROCKY Dataset.

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 Lat-Lon 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 Lat-Lon 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 with GNSS signal

and without 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,
  - 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.

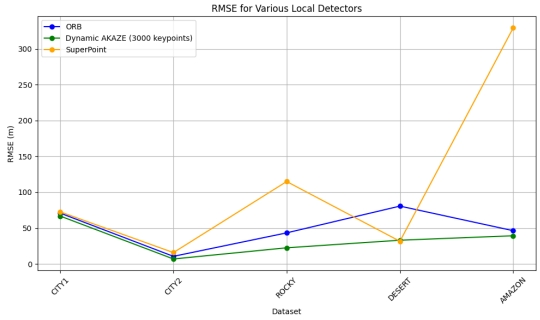


Figure 4.6: Radial Error for Various Local Detectors.

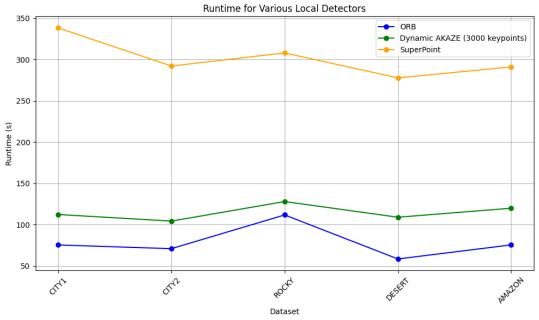


Figure 4.7: Runtime for Various Local Detectors.

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 Lat-Lon 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, BFMatcher's runtime was significantly higher than FLANN's. This is attributed to FLANN's approximate matching, which scales better with the number of keypoints.

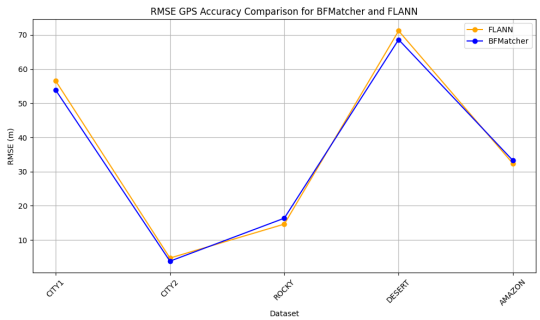


Figure 4.9: Radial Error for BFMatcher and FLANN.

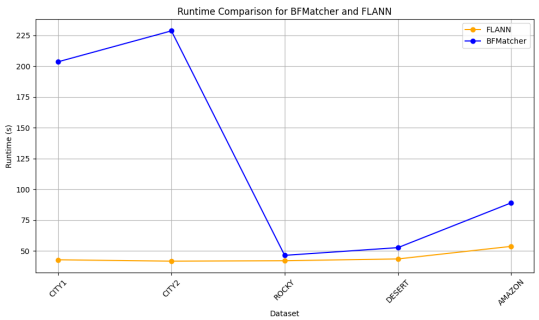


Figure 4.10: Runtime Comparison for BFMatcher and FLANN.

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 Lat-Lon 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.

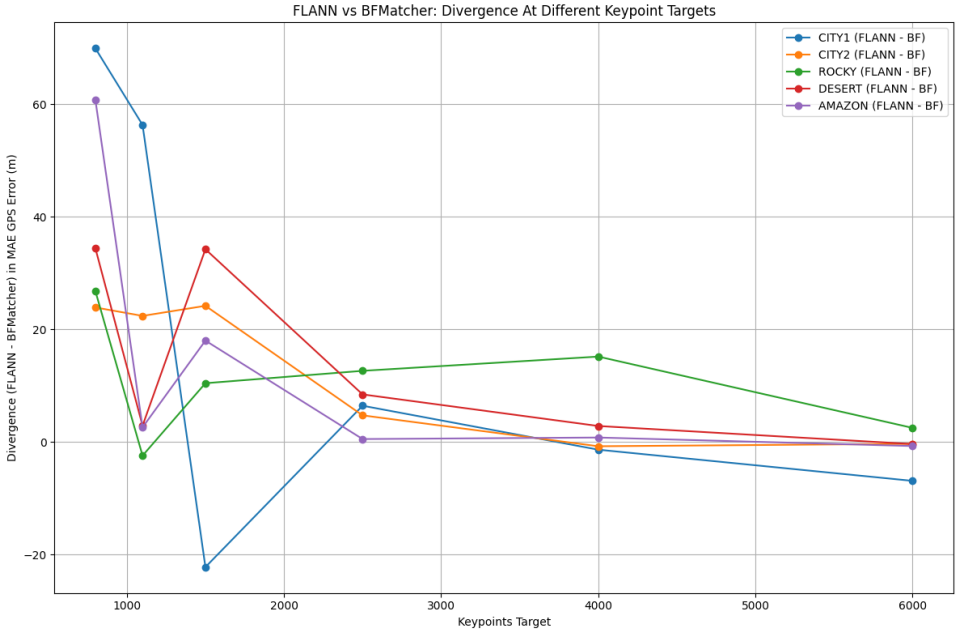


Figure 4.11: Convergence in RMSE Lat-Lon Error Between FLANN and BFMatcher Across Keypoint Targets.

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.

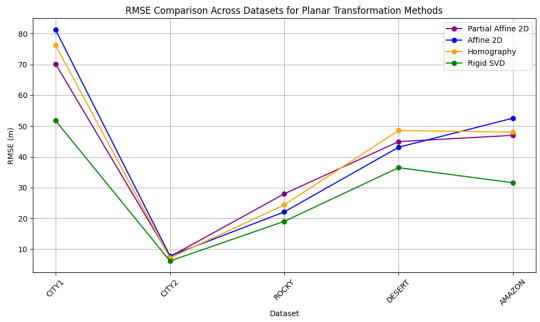


Figure 4.12: RMSE Comparison Across Datasets for Planar Transform Estimators.

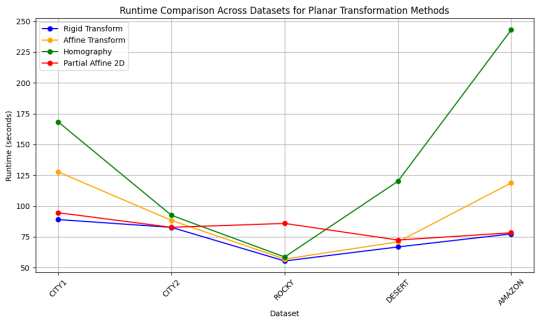


Figure 4.13: Runtime Comparison Across Datasets for Planar Transform Estimators.

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 RANSAC, or less often, LMEDS or other outlier rejection methods for robust outlier rejection. These thresholds 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. Thus, OpenCV methods achieve a score of 3 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 Lat-Lon 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 Lat-Lon 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 and Cross-Correlation technique perform near equivalently, with SSIM marginally behind. 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 sufficient 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 most consistent runtimes, followed closely by Cross Correlation. SSIM exhibited the longest runtimes of the remaining methods due to its more complex structural similarity calculations.

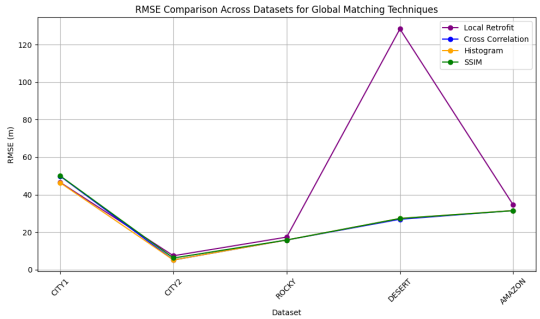


Figure 4.15: RMSE Comparison Across Datasets for Global Matching Techniques.

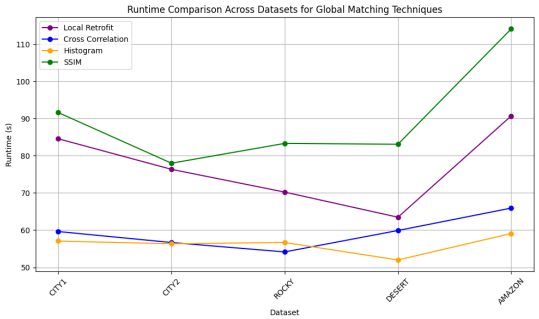


Figure 4.16: Runtime Comparison Across Datasets for Global Matching Techniques.

4.5.2. Robustness Testing

The global matching methods did not have any tunable parameters, achieving them a robustness score of 5. However, the local retrofit model had various parameters that could be tuned, including detector type, detector threshold, grid size, match threshold. Further, the runtime and accuracy were extremely difficult to balance, requiring extremely precise tuning to achieve usable performance. As such, the local retrofit model achieved a robustness score of 2.

4.5.3. Considerations for Alignment Prior to GLobal Matching

This sections details the considerations when choosing the precision of the rotational alignment techniques employed to align the image pairs for unbiased similarity comparisons between the images. This test subjects the estimated internal alignment to an additional skew and notes their response in best match choice, realized through radial error. This was first tested under a 5-degree skew, where no changes occur. Figure 4.17 shows the percentage change in Lat-Lon error from the prior estimate, with no skew, with a 10-degree skew. Cross-correlation maintains the most similar accuracy, closely followed by histogram and then SSIM. All methods are highly robust to rotational misalignments up to 5 degrees, however, should not exceed this. As such, when employing a rotational alignment technique prior to global matching, a 5-degree skew is tolerable and the choice of detector and matcher may be guided by efficiency rather than accuracy.

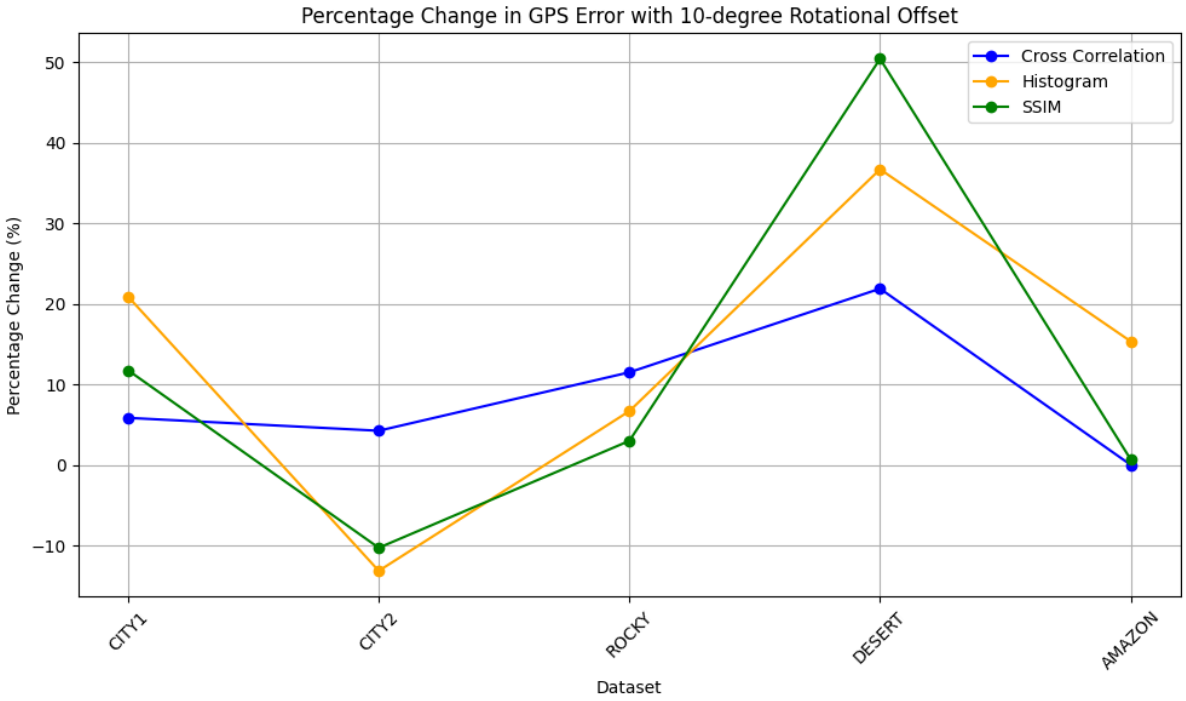


Figure 4.17: Percentage Change in Lat-Lon Error with 10-degree Rotational Offset

4.5.4. 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

Various methods were 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 radial error. LMEDS was also significantly faster than RANSAC, making it the preferred choice for planar transform outlier rejection. The results are summarized in Figure 4.18 and Figure 4.19. Both thresholds require precise tuning, with LMEDS requiring slightly less tuning. Both methods achieved a robustness score of 3, as they required precise tuning to achieve optimal performance across datasets.

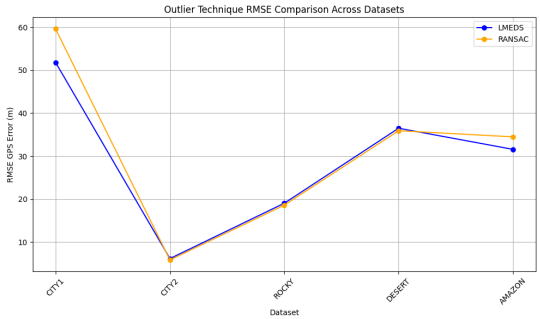


Figure 4.18: Radial Lat-Lon RMSE Comparison Across Datasets for LMEDS and RANSAC.

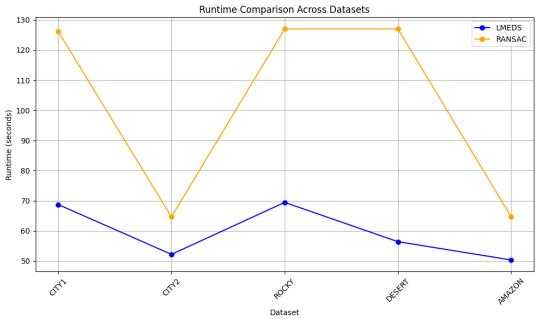


Figure 4.19: Runtime Comparison Across Datasets for LMEDS and RANSAC.

4.6.2. Lowe's Ratio Test

Lowe's Ratio Test was utilized to filter keypoint matches by comparing the distance of the best match to that of the second-best match. Initially, a static threshold was applied to determine match quality. However, this static approach was insufficient for handling the variability across diverse datasets, resulting in inconsistent accuracy. To improve robustness, a dynamic thresholding strategy was adopted. This approach involves setting an initial threshold and incrementally increasing it, thereby allowing greater leniency until a predefined number of matches or percentage of the keypoints is found.

The effectiveness of this dynamic thresholding method was evaluated through empirical testing. A lower initial threshold and lower increment value increased the likelihood of approaching the desired match count but impacted runtime. The initial threshold was

set to 0.7, and the increment was established at 0.05. These parameters were chosen because they provided a balance between efficiency and accuracy; it may be changed crudely around these values without a significant impact on runtime or accuracy.

Through testing, a target of 500 keypoints was selected for its ability to consistently maintain stability across diverse datasets. The optimal range for reliable performance was found to be 300 to 700 keypoints, with 500 keypoints delivering the most stable results. Additionally, to prevent low-quality matches from entering the system when fewer keypoints were detected, a maximum match-to-keypoint ratio of 75% was implemented. Given that the prior stage, Absolute Thresholding, was capped at, and expected to obtain, 1000 keypoints, this ratio typically exceeds the 500 match target, underscoring its role as a redundancy measure.

4.6.3. N-Match or Absolute Thresholding

Absolute Thresholding, or N-Match Thresholding, involves filtering keypoint matches based on a fixed number of matches or a specific descriptor distance threshold. This method was tested in two configurations: applied before Lowe's Ratio Test and applied after.

When Thresholding was applied after Lowe's Ratio Test, it operated on an already filtered set of matches. Therefore, this configuration proved to be highly sensitive to the chosen threshold. Empirical tests demonstrated that slight adjustments to the threshold often subtended situations of negligible effect or removal of almost all matches, leading to unpredictable performance and inconsistent accuracy across different datasets.

In contrast, applying Absolute Thresholding before Lowe's Ratio Test offered significant advantages. This pre-filtering step effectively reduced the number of matches early in the processing pipeline, thereby decreasing the computational load for subsequent stages. Feature detectors occasionally produced abnormally high keypoint counts, up to five times the expected number of keypoints, due to variations within datasets, which led to excessive computational costs in Lowe's Ratio Test and subsequent transformation estimation stages, as well as an increase in noisy matches. By limiting the number of keypoints through Absolute Thresholding, these outliers were removed, enhancing both efficiency and match quality.

N-Match thresholding was chosen above descriptor based thresholding due to its simple adaptability across terrains with varying quality keypoints. During the testing phase, various match threshold values between 1000 and 2500 keypoints were evaluated to identify the optimal balance between runtime efficiency and matching accuracy. Thresholds well below this keypoints were found to be too restrictive, while those above the limit had negligible effect. A threshold of 1000 keypoints was chosen due to significant gains in runtime relative to the upper limit, with negligible effect on accuracy. Naturally, this was

also chosen in conjunction with Lowe's ratio's target match count of 500, to ensure that it had sufficient keypoints to meet this requirement.

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 similarity alignment layer due to its efficiency, and Dynamic AKAZE for the dense transformation estimation layer due to its accuracy.
- **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 accuracy and runtime.
- **Optimization Techniques:** N-match thresholding targeting 1000 keypoints, Lowe's filtering targeting 500 keypoints. This was chosen to balance quality and quantity of matches. LMEDS or RANSAC was not included due to prior choices of methods not requiring it.

Chapter 5

Results

XXX - CV needs pictures. look at 5.5 pixels and say why that bad, visualize maybe single image for match not side by side. - not clear what its finding, but it is robust which is nice - try ask what are these dots... what are keypoints - GitHub link for code. Mentionthis is how it was implemented, not flow diagram. So when someone goes on Github they know whats going on. Crop Earth overlap relative to my baseline image 972. no 1080. CTRL F 1080. xxx - show matches black and white. / xxx show image overlay.

5.1. Key Metric 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. More information about the testing setup is provided in 4

5.1.1. Key Performance Metrics

The system's performance across the five diverse datasets—CITY1, CITY2, ROCKY, DESERT, and AMAZON—is evaluated based on Accuracy and Runtime.

Accuracy is assessed in two primary ways:

1. *Absolute Radial Error (RMSE)*: Measured in the estimates radial metres from the ground truth, averaged for the dataset. This metric provides a tangible understanding of the error magnitude and the system's responsiveness to movement size. Additionally, it may be represented per-image, in pixels or in axial components, offering an intuition of error size relative to the fixed resolution of 1920x972 pixels.

2. *Relative Radial Error*: Expressed as the per image Absolute Radial Error as a percentage of the ground truth radial change in distance, averaged for the dataset. The ground truth radial change is the magnitude of the translation vector from the reference image's center to the current image's center, both measured in meters. This normalized metric allows for a clearer understanding of the system's accuracy relative to the movement size. It may also be represented per-image or in axial components.

Runtime is evaluated in three stages: 1. *Mean Add Time (With GNSS)*: The average time required to add one image to the pipeline, including streaming the image and extracting its features. This time determines the system's processing rate while GNSS signals are available after parameter inference has concluded.

2. *Mean Parameter Inference Time (With GNSS)*: The average time needed to infer the pixel-to-meter conversion factor per image. This includes processing the entire pipeline (finding the best match and subsequent transformation estimation) to estimate the UAV's position, and computing linear regression by leveraging the ground truth once at the end of the mode. Adding this to the add time provides the total time to process an image when GNSS is available and parameter inference is active. Inference mode is active for the first five images, since further tests continued to limit the reference image capture rate while producing negligible improvements in accuracy.

3. *Mean Location Inference Time (Without GNSS)*: The average time taken to infer the UAV's location when GNSS signals are lost. Measured per image, this time encompasses processing the entire pipeline to estimate the UAV's position. This metric is critical as it determines how quickly a pilot can correct any path deviations to prevent further drift.

The results show that the system achieves consistent accuracy and runtime performance across each dataset. This section details the performance metrics for each dataset, while the following sections analyze the sources of error where applicable.

In terms of accuracy, all datasets achieved a mean radial error below 1% of the ground truth distance, a mean pixel error being below 1.05 pixels a mean meter error below 3.2 meters. As the percentage error of the ground truth distance is the most fair metric, the order of average performance is therewith evaluated. The CITY2 dataset showed the best performance, followed by the DESERT dataset, then the AMAZON dataset, then the CITY1 dataset, and finally, with a large difference in performance, the ROCKY dataset.

In terms of maximum single image errors across datasets, the maximum error ocurred in the ROCKY dataset. This maximum radial error 6.21% of the ground truth distance, which corresponds to 27.54 meters or 4.93 pixels. In terms of maximum percentage error, the CITY2 dataset showed the best performance, followed by the DESERT dataset, then the CITY1 dataset, then the AMAZON datast, and finally, with a large difference in performance, the ROCKY dataset.

In terms of mean runtimes, the mean add time across datasets was below 0.6 seconds, the mean parameter inference time was below 1.4 seconds, and the mean location inference time was below 1.3 seconds. This indicates that the system is on average able to maintain real-time performance while inference mode is on, with the sum of the parameter inference and add times being below 2 seconds. Further, the the location inference time is below 2 seconds.

However, when looking at maximum runtime, the system produced runtimes over 2 seconds while inference mode was on. While off, the system maintained real-time

performance. This means that real-time performance was not met while inference mode was on. However, inference mode is only applied to the first few images, meaning the time may be caught up relatively quickly while inference mode is off, as images will be stored prior to their usage in the pipeline. This basically means if the system is behind at some point, the images are still there for later catchup.

Table 5.1: Mean Radial Errors

	CITY1	CITY2	ROCKY	DESERT	AMAZON
Percentage Radial Error (%)	0.4001	0.2290	1.9154	0.2515	0.3811
Pixel Error	0.8196	0.6114	1.0520	0.5668	0.3919
Meter Error	4.7978	2.7576	5.8822	2.3573	2.5336

Table 5.2: Maximum Radial Errors

	CITY1	CITY2	ROCKY	DESERT	AMAZON
Percentage Radial Error (%)	0.6356	0.2715	6.2123	0.3516	0.7447
Pixel Error	2.2948	0.9949	4.9280	1.7829	1.0508
Meter Error	13.4077	4.4886	27.5392	7.4109	6.7927

Table 5.3: Mean Processing Times (Seconds)

	CITY1	CITY2	ROCKY	DESERT	AMAZON
Add Time	0.5603	0.5454	0.4912	0.4171	0.5314
Parameter Inference Time	1.3363	1.2972	1.2658	1.3197	1.2271
Location Inference Time	1.2028	1.2866	1.1829	1.2862	1.2837

Table 5.4: Maximum Processing Times (Seconds)

	CITY1	CITY2	ROCKY	DESERT	AMAZON
Add Time	0.7671	0.6438	0.6098	0.5649	0.6731
Parameter Inference Time	1.2706	1.4904	1.6314	1.6769	1.7244
Location Inference Time	1.3609	1.6271	1.4374	1.8392	1.7091

5.1.2. Flight Path Analysis

Figures 5.1 and 5.2 show the actual vs estimated flight of the UAV across the worst and best performing dataset, ROCKY and DESERT respectively. Even in the worst performing dataset, the system maintains a highly accurate flight path. On a global scale, this error is negligible. The pilot be able to control the UAV and prevent further drift under significantly worse conditions.

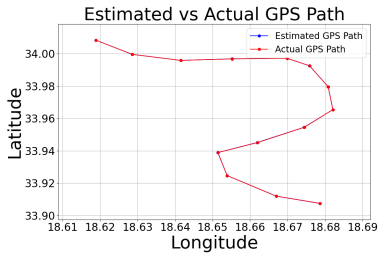


Figure 5.1: Flight Path of UAV in Rocky Dataset (Worst)

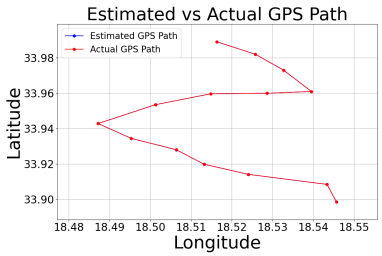


Figure 5.2: Flight Path of UAV in Desert Dataset (Best).

5.1.3. Error Distribution Map

The heatmap in Figure 5.3 illustrates the distribution of pixel deviations in the X and Y directions across the datasets. For spatial conciseness, some axial values are omitted, and the values are quantized into bins, meaning they are not exactly off by the pixel bin they are in. This visualization provides an intuitive understanding of the pixel error distribution and any potential axial bias. As shown, the errors are distributed relatively evenly across both axes, indicating no significant axial bias in the system. Moreover, the majority of errors are below 2 pixels, demonstrating the system's robustness and accuracy in localization estimates. The largest radial error, as noted in Table **??**, is 4.9199 pixels. While this error is relatively small, the sources of these errors are discussed in the following section.

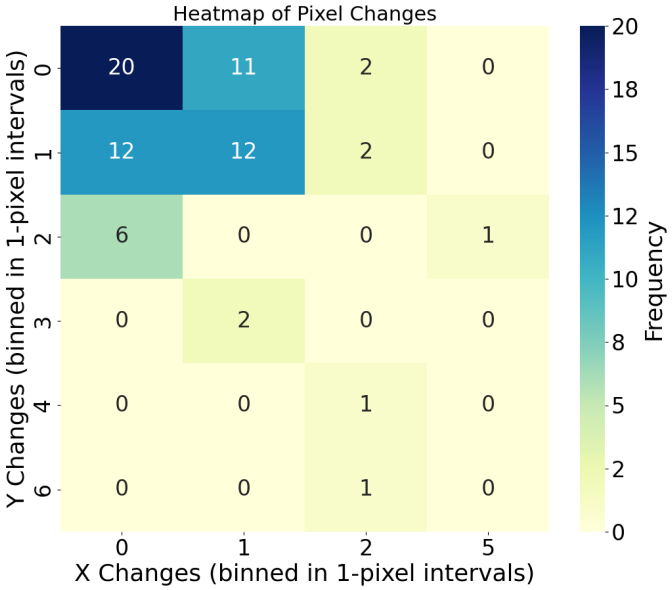


Figure 5.3: Heatmap of Pixel Deviations in X and Y Directions

Sources of Error

xxx The system has a maximum error of 6.2% on the ROCKY dataset. Further, it has relative outliers in the CITY1 and AMAZON datasets of a maximum error of 0.64% and 0.74%. Sources of these errors are detailed below, with these effects being common to all datasets but especially prevalent in some.

Quantization: When images are taken from slightly different positions, feature edges can distort due to sub-pixel shifting. This is because a fixed resolution can averages any information smaller than a pixel. For large features, these distortions at feature boundaries are minimal relative to feature size, but for smaller features, edge distortions occupy a significant portion, leading to high levels of distortion. At high altitudes, where features are predominantly small, this effect distorts a large number of descriptors and keypoints, leading to considerable distortion.

This issue is prominent in the CITY datasets, where dense, small features amplify sub-pixel errors, leading to higher overall errors. While CITY2's dataset also experiences sub-pixel shifting, it involves only translations between frames rather than rotations too, which has a larger impact on reducing estimation error overall; hence, the error is not explicitly visible in the CITY2 dataset.

Depth and Perspective Changes: Google Earth uses a 3D model to generate its terrain, leading to changes in perspective and ground height as images are taken from different spots in the landscape. This effect is amplified under presence of larger changes in altitude in the terrain.

This effect is particularly evident in the ROCKY dataset, where large height changes in the mountainous regions cause both perspective distortion and scale changes.

Homogenous, Repetitive terrain: The ability for a detector to find keypoints depends on the number of highly unique, high contrast features are present in the environment. Additionally, the matcher's ability to discern the correct corresponding keypoint from a large array thereof depends on whether the local environment of the feature, or its descriptor, is significantly different to that of the other keypoints. In other words, the more homogenous and repetitive an environment is, the less likely a feature detector is to find many good features and a matcher is to choose the correct matches respectively.

The strengths of modern Computer Vision techniques allows this effect to be almost entirely negated. However, the most homogenous and repetitive environment, the AMAZON desert, is marginally affected by this.

Variations in Terrain and Environmental Conditions: Each terrain in the dataset has unique characteristics, such as differences in good feature count and uniqueness of features. The optimal number of good features varies with the terrain. In this study, a static target number of keypoints and matches was applied to each stage, which does not fully account for these differences. The system may generalize well across multiple

datasets but it is not perfectly optimized for any specific one.

Algorithmic Constraints: The selected image processing and matching algorithms have inherent limitations, especially under extreme conditions or with highly repetitive features. Additionally, being optimized for efficiency, they do not capture and represent every feature within an image perfectly.

5.2. Stress Testing

To test whether this system is truly viable in practice, it needs to be tested under a variety of non-optimal conditions. In this section, tests are conducted under low-light modes, dynamically varying lighting conditions, low resolution, and low reference image overlap (the number of mutual pixels). To maintain scope, larger practical limitations are not tested, such as extremely dynamic environments, like city lights, and large occlusions, such as clouds. 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. XXX.

5.2.1. Low Resolution Testing

The table below shows the system's performance under varying resolution downscaling factors, referenced with the output resolution. The system's accuracy and runtime are analyzed as the resolution decreases, providing insights into the trade-off between navigational accuracy and computational efficiency. The results aim to identify the minimum resolution required for reliable navigation and real-time performance.

As the resolution decreases, there is hardly any change in performance until a very sharp increase in error at 432 x 218.7. Further, the method does not find any keypoints below the minimum shown resolution. However, the ability of the method to work optimally at around 480x243 pixels indicates the ability for the system to work xxx other datasets.

Resolution (Width x Height)	Mean Radial Error (%)	Mean Location Inference Time
1920 x 972	0.2218	1.9917
1536 x 777.6	0.2175	1.8267
1152 x 583.2	0.2227	1.0204
768 x 388.8	0.2413	0.8008
576 x 291.6	0.2163	0.5876
528 x 267.3	0.3262	0.7108
480 x 243	0.2412	0.6379
432 x 218.7	3.8517	0.4977
384 x 194.4	64.6572	0.5021

Table 5.5: Resolution vs Mean Radial Percent and Mean Location Inference Time

Resolution	Mean Radial Percent (%)	Mean Location Inference Time (s)
1920, 972	0.4218	1.5190
1536, 777.6	0.4283	1.6812
768, 388.8	0.4506	0.8300
576, 291.6	0.5819	0.4359
528, 347.49	6.8508	0.4015
480, 315.9	8.0347	0.4110
432, 284.31	11.9210	0.3779
384, 252.72	22.2008	0.3591
336, 221.13	29.2928	0.3452

Table 5.6: DATSETROT2: Resolution vs Mean Radial Percent and Mean Location Inference Time

Resolution	Mean Radial Percent (%)	Mean Location Inference Time (s)
1920, 972	0.2216	1.7689
1536, 777.6	0.2298	1.6480
768, 388.8	0.2393	0.6818
576, 291.6	0.2505	0.4863
528, 347.49	12.1395	0.4111
480, 315.9	12.1405	0.3987
432, 284.31	12.2489	0.3669
384, 252.72	14.5228	0.3273
336, 221.13	20.0428	0.3367

Table 5.7: DATSETCPT2: Resolution vs Mean Radial Percent and Mean Location Inference Time

Resolution	Mean Radial Percent (%)	Mean Location Inference Time (s)
1920, 972	80.2173	2.0976
1536, 777.6	1.4441	1.8400
768, 388.8	1.6923	0.7002
576, 291.6	1.8616	0.7214
528, 267.3	2.0240	0.4917
480, 315.9	51.6151	0.3572
432, 284.31	46.3796	0.3670
384, 252.72	51.2201	0.3496
336, 221.13	73.3333	0.3020

Table 5.8: DATSETROCK2: Resolution vs Mean Radial Percent and Mean Location Inference Time

Resolution	Mean Radial Percent (%)	Mean Location Inference Time (s)
1920, 972	20.8159	2.1469
1536, 777.6	0.2945	1.5383
768, 388.8	0.3325	0.6885
576, 291.6	0.2974	0.3736
528, 347.49	13.2917	0.3159
480, 315.9	12.2575	0.2913
432, 284.31	24.0492	0.2324
384, 252.72	14.5317	0.2088
336, 221.13	26.0281	0.2179

Table 5.9: DATSETSAND2: Resolution vs Mean Radial Percent and Mean Location Inference Time

Resolution	Mean Radial Percent (%)	Mean Location Inference Time (s)
1920, 972	30.3264	2.1608
1536, 777.6	18.2228	1.8499
768, 388.8	0.6585	0.6561
576, 291.6	18.0415	0.3592
528, 347.49	24.7928	0.3876
480, 315.9	35.0960	0.3423
432, 284.31	69.1683	0.3214
384, 252.72	82.1270	0.3068
336, 221.13	354.0224	0.3091

Table 5.10: DATSETAMAZ2: Resolution vs Mean Radial Percent and Mean Location Inference Time

5.2.2. Dynamic Lighting Testing

This test aims to evaluate the system's robustness under varying lighting conditions. During long missions, the UAV may only return along the outbound path at a significantly later time, where lighting conditions have significantly changed. To simulate the varying extremities, a random number generator decides on the time of day setting. Note, time of day in this context purely refers to a lack of light conditions, where contrast, brightness, and tint are varied as well as noise levels. However, these conditions do not account for other typical occurrences, such as city lights. Since the image similarity estimator will tend to account for these variations in lighting and choose the least affected image, the lighting variations are only applied after the best match has been chosen. In reality, the lighting conditions of potential reference images in the proximity will be similar, and the image similarity estimator will not have the choice between images that are highly different in lighting conditions.

5.2.3. Mutual Information Results

This section evaluates the system's performance under reduced mutual information, which is the number of pixels shared between images, as a percentage of the image size. The system's accuracy and runtime are analyzed as the mutual information decreases, providing insights into the trade-off between navigational accuracy and computational efficiency. The results aim to identify the minimum mutual information required for reliable navigation and real-time performance.

Methodology

To emulate a decrease in mutual information, images are progressively cropped and the mutual information is calculated for each instance. The CITY2 dataset is utilized, focusing exclusively on translational changes which simplifies the mutual pixel calculation to the difference between the image size and translation vector. Cropping is performed at a rate of 50 pixels per iteration, scaled up in the x direction by the aspect ratio, alternating between x and y crops. Since translation vectors vary between images, only the mutual information of the bottleneck image (the one with the lowest mutual information) is shown for each iteration to maintain visual brevity. xxx - reference images The accuracy plot indicates that the error remains almost constant until the overlap drops below approximately 30%, at which point the error starts to increase. It is important to note that below the lowest tested value, the mutual information of the bottleneck image was below 0, hence the error from thereon is not shown. Interestingly, the error decreases again after a significant increase at an even lower overlap. However, this reduction is unreliable and results from coincidental reductions in non-mutual information causing a high density of good matches, despite the

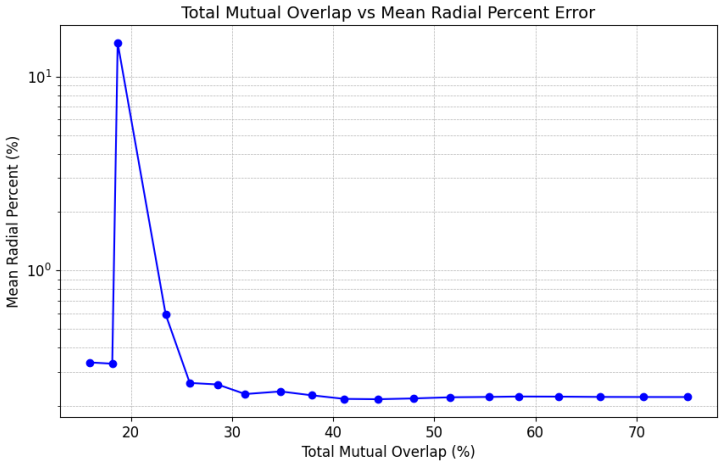


Figure 5.4: Mutual Information vs Mean Error Percentage

low number of mutual pixels. Therefore, extremely low levels of mutual information cannot be relied upon for consistent accuracy. Maintaining an overlap above 30% ensures stable and accurate navigation. In practice, where possible, the UAV should aim to maintain view of at least 50% of the path to err on the side of caution.

The runtime plot shows a underlying linear trend in runtime and mutual information. As expected, the runtime decreases with a decrease in mutual information, as the number of features and matches decreases with the image size, reducing the computational load. This rate is slightly above 0.1 s per 10% reduction in mutual information. This decrease is moderate and measures may be taken to decrease the field of view of the camera when the UAV is close to the path, to reduce the computational load.

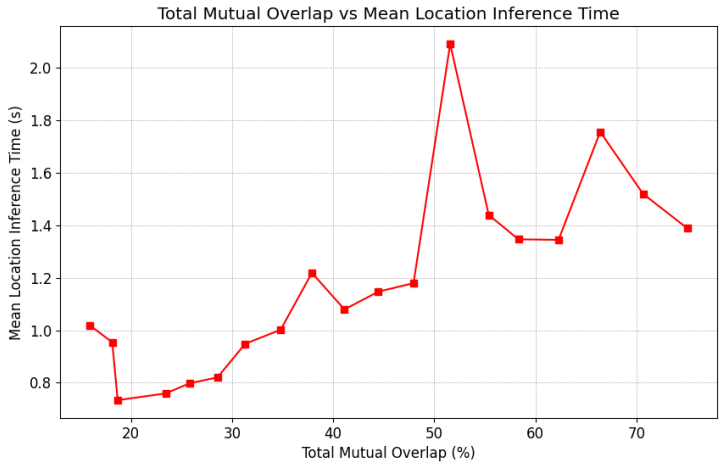


Figure 5.5: Mutual Information vs Mean Localization Time

5.2.4. Low-Light Testing

In the context of UAV navigation, long missions may require the UAV to return along the outbound path at a significantly different time to the outbound journey. In this test, the reference image lighting conditions are altered relative to that of the current image. This adds an extra layer of difficulty to the task above simply changing all of the images to low-light would not fully capture; however, the ability to operate under low-light conditions is implicitly tested here too. Since darker images have less uniqueness and contrast, the number of features found will become too low if the dynamic detector threshold is adjusted to target a specific number of keypoints for day time conditions. Therefore, it is adjusted for the simulated night-time conditions, acknowledging this more lenient threshold leads to more keypoints and subsequently higher runtime. In this study, the reference images are set to night-time conditions, whilst the current images are set to day time conditions. In practice, longer missions need to consider the length of the mission and the time of day the UAV is expected to return.

This study evaluates the robustness of the navigation method under simulated evening and nighttime conditions across five datasets: CITY1, CITY2, ROCKY, DESERT, and AMAZON. The following parameters visible in table 5.11 are used to simulate the lighting effects for the different conditions, with day applying no effects. The alpha parameter controls contrast to adjust how defined or soft the image looks. Beta darkens the image to simulate various levels of night. Noise sigma adds Gaussian noise to mimic low-light imperfections. Weight balances the processed image and Gaussian noise effect for realism in night simulations, with higher values maintaining clarity and lower adding more distortion.

Effect	Alpha (Contrast)	Beta (Brightness)	Noise Sigma	Weight
Early Evening	0.7	-15	10	0.97 main, 0.03 noise
Late Evening	0.6	-30	15	0.95 main, 0.05 noise
Late Night	0.5	-50	20	0.92 main, 0.08 noise

Table 5.11: Parameter Settings for Different Lighting Effects

The alpha parameter controls contrast to adjust how defined or soft the image looks. Beta darkens the image to simulate various levels of night. Noise sigma adds Gaussian noise to mimic low-light imperfections. Weight balances the processed image and noise for realism in night simulations, with higher values maintaining clarity and lower adding more distortion.

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 5.6: Daytime Image of Cape Town

Figure 5.7: Early Evening Image of Cape Town

Figure 5.8: Late Evening Image of Cape Town

Figure 5.9: Night Image of Cape Town

Figure 5.10: Lighting Conditions in Cape Town

Results

Here's a polished version of the paragraph:

Figure 5.11 illustrates the percentage increase in error, compared to daytime conditions, of the navigation method under early evening, late evening, and nighttime settings for each dataset.

The CITY1 dataset shows the smallest increase in error across all conditions, maintaining an error increase under 10%. CITY2 follows with slightly higher increases in error as lighting conditions become more challenging. This could be attributed to the dataset's minimal rotations, which do not implicitly remove non-mutual information and thereby reduce false positives under difficult conditions, as is the case with the CITY1 dataset. The strong performance of the CITY datasets can be attributed to their dense feature environments with numerous high-contrast elements. However, in real-world scenarios, city lights—unaccounted for due to dataset limitations—could introduce dynamic changes impacting accuracy, especially if they occupy a significant portion of the view.

The ROCKY dataset ranks third in performance, with the DESERT and AMAZON datasets experiencing the largest increases in error. The substantial error rise in these datasets is due to their already sparse environments becoming even more feature-poor under low-light conditions, significantly increasing navigation error. The midnight and late evening settings are particularly challenging, with errors exceeding 100 times the baseline daytime error in these sparse environments.

Despite the high relative error percentages, baseline errors are typically below 1% for most datasets, with only ROCKY exceeding 2%. This means that, under early evening conditions, the relative error remains below 5% for all datasets. In denser environments like the CITY datasets, errors remain under 2% across all lighting conditions. For sparser environments, it is important to mitigate illumination changes during image capture, although the pipeline demonstrates robustness to these variations.

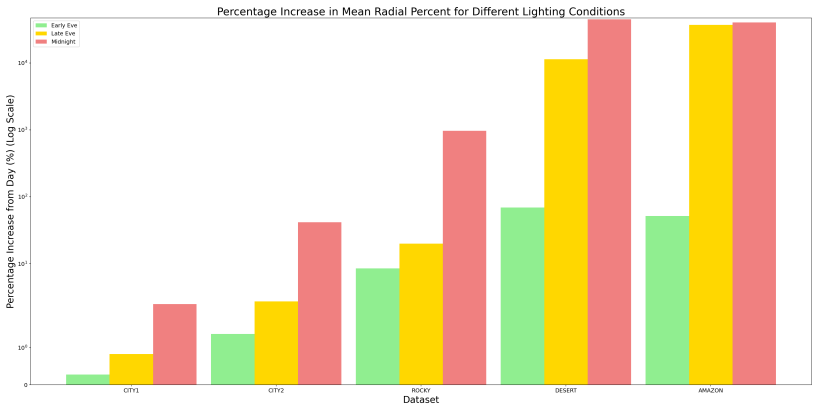


Figure 5.11: Mean Radian Increase in Error Under Nighttime and Evening Conditions.

Dataset	Day (%)	Early Eve (%)	Late Eve (%)	Midnight (%)
CITY1	0.362	0.363	0.365	0.371
CITY2	0.219	0.222	0.225	0.310
ROCKY	1.909	2.071	2.290	20.399
DESERT	0.251	0.425	28.713	112.610
AMAZON	0.381	0.578	141.647	153.678

Table 5.12: Mean Radial Percentages for Different Datasets under Varying Lighting Conditions

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 consistent and strong accuracy and runtime performance across various datasets, including sparse-feature environments, showcasing its ability to generalize effectively. Despite large variations in the terrains as well as other practical limitations faced, the pipeline demonstrated remarkable resilience. Further, in reality, once the path is found, there will be significantly less translation between the current and reference images, leading to a lower error ceteris paribus.

Further, the applied stress tests showed the pipelines invariance to

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] J. Khalil, "Gnss spoofing threatens airline safety, alarming pilots and aviation officials," *GPS World*, 2024. [Online]. Available: https://www.gpsworld.com/gnss-spoofing-threatens-airline-safety-alarming-pilots-and-aviation-officials/

[6] 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.

[7] 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.

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

[9] 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/

[10] M. Y. Arafat, M. M. Alam, and S. Moh, "Vision-based navigation techniques for unmanned aerial vehicles: Review and challenges," *Drones*, vol. 7, no. 2, p. 89, 2023, accessed: 2024-10-29. [Online]. Available: https://doi.org/10.3390/drones7020089

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

[14] 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

[15] 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

[16] 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

[17] 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

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

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

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

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

[22] 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

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

[24] 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

[25] 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

[26] 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/

[27] ——, "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/

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

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

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

[31] 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/

[32] 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.