5-2018

Homography-Based State Estimation for Autonomous Exploration in Unknown Environments

Andres S. Chavez Armijos

HOMOGRAPHY-BASED STATE ESTIMATION

FOR AUTONOMOUS EXPLORATION IN UNKNOWN ENVIRONMENTS

A Thesis

Submitted to the Faculty

of

Embry-Riddle Aeronautical University

by

Andres S. Chavez Armijos

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aerospace Engineering

May 2018

Embry-Riddle Aeronautical University

Daytona Beach, Florida

HOMOGRAPHY-BASED STATE ESTIMATION
FOR AUTONOMOUS EXPLORATION IN UNKNOWN ENVIRONMENTS

by

Andres S. Chavez Armijos

A Thesis prepared under the direction of the candidate's committee chairman, Dr. Richard Prazenica, Department of Aerospace Engineering, and has been approved by the members of the thesis committee. It was submitted to the School of Graduate Studies and Research and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chairman, Dr. Richard Prazenica

Member, Dr. Hever Moncayo

Member, Dr. Troy Henderson

Graduate Program Coordinator, Dr. Magdy Attia — 4.23.2018 Date

Dean of College of Engineering, Dr. Maj Mirmirani — 04/23/18 Date

Vice Chancellor, Academic Support, Dr. Christopher Grant — 4/25/18 Date

ACKNOWLEDGMENTS

I would like to dedicate this thesis to all the people that have supported my work in a directly or indirectly manner, but I would also like to thank all of those who have meant something during my graduate studies.

Firstly, I would like to thank my committee members for their support on this thesis, but I would also like to give a special mention to Dr. Richard Prazenica, who has not only been a good mentor and advisor, but has also been a truly good friend that I will always remember. Secondly, I would like to thank my friend Nick Myhre who has aided me through the process of debugging, mathematical reasoning, and flight testing behind this thesis. He has also been a friend that I can truly rely on for the next chapters of my life. I would also like to thank my friends and roommates Erick, Sergio, and Carlos who were an important part of the technical and emotional support that I received during my thesis.

Lastly, but not least, as Jane Horward once said ”Call it a clan, call it a network, call it a tribe, call it a family. Whatever you call it, whoever you are, you need one”, this thesis would not have been possible without the support of my family. Therefore, this work is specially dedicated to all of the members of my family with special mention to Alexandra, Jose, Mauricio, and Rosa because this work is the reflection of all the effort that they made so I could follow my dreams. They have always been and always will be my motivation, inspiration, and my source of strength for every piece of my life.

Gracias familia por la oportunidad que me dieron. Los amo.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

SYMBOLS

$\vec{a}^B$	Acceleration vector in body reference frame
$\vec{\omega}^B$	Angular velocity vector in body reference frame frame
ρ	Euler angles
f	Focal length
g	Gravity constant
H	Homography matrix
J_m	Inertia matrix
K_{int}	Camera intrinsic matrix
$\vec{m}^B_{mag}$	Magnetic field vector in body reference frame
Ψ_{mag}	Magnetic heading
m	Mass
$\vec{N}$	Normal vector
P^N	Position vector in navigation reference frame
R^b_a	Rotation matrix from a frame to b frame
t	Time
τ	Torque
$T^a_{a \to b}$	Translation from a to b with respect to frame a
$vec\{\}$	Vector operator
V^N	Velocity vector in navigation reference frame

ABBREVIATIONS

ABA	Appearance-Based Approach
ACC	Accelerometers
AHRS	Attitude and Heading Reference System
BARO	Barometer
CG	Center of Gravity
DCM	Direction Cosine Matrix
DLT	Direct Linear Transformation
DoF	Degrees of Freedom
DR	Dead Reckoning
EKF	Extended Kalman Filter
FFI	Federated Filtered Integration
GPS	Global Positioning System
GNSS	Global Navigation Satellite Systems
GYR	Gyroscope
IF	Innovation Filtering
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KLT	Lucas-Kanade Tracker
LiDAR	Light Detection and Ranging
LT	Landing Target
MAG	Magnetometer
MAV	Micro Aerial Vehicle
MBA	Model-Based Approach
MEMS	Micro Electro Mechanical Systems
MF	Median Filtering
NED	North East Down
RANSAC	Random Sample Consensus
RMS	Root Mean Square
SE	Single Epoch
SLAM	Simultaneous Localization and Mapping
SURF	Speeded-Up Robust Features
TRN	Terrain-Referenced Navigation
UAS	Unmanned Autonomous System
UAV	Unmanned Autonomous Vehicle
UKF	Unscented Kalman Filter
VBN	Vision-Based Navigation
VO	Vission-Based Odometry

ABSTRACT

Chavez Armijos, Andres S. MSAE, Embry-Riddle Aeronautical University, May 2018. Homography-Based State Estimation for Autonomous Exploration in Unknown Environments.

This thesis presents the development of vision-based state estimation algorithms to enable a quadcopter UAV to navigate and explore a previously unknown GPS denied environment. These state estimation algorithms are based on tracked Speeded-Up Robust Features (SURF) points and the homography relationship that relates the camera motion to the locations of tracked planar feature points in the image plane. An extended Kalman filter implementation is developed to perform sensor fusion using measurements from an onboard inertial measurement unit (accelerometers and rate gyros) with vision-based measurements derived from the homography relationship. Therefore, the measurement update in the filter requires the processing of images from a monocular camera to detect and track planar feature points followed by the computation of homography parameters. The state estimation algorithms are designed to be independent of GPS since GPS can be unreliable or unavailable in many operational environments of interest such as urban environments. The state estimation algorithms are implemented using simulated data from a quadcopter UAV and then tested using post processed video and IMU data from flights of an autonomous quadcopter. The homography-based state estimation algorithm was effective, but accumulates drift errors over time due to the relativistic homography measurement of position.

1. Introduction

1.1 Problem Statement

Recent efforts have been made towards the development of new unmanned autonomous systems (UAS) capable of performing missions in environments that are difficult for humans to access. Therefore, such systems can be seen in several applications ranging from space exploration, mineral rock sampling, agricultural irrigation, and surveillance and reconnaissance for military applications. The success of such applications relies on the development and implementation of more complex and intelligent systems capable of making decisions autonomously. Most successful onboard systems are based on the usage of inertial measurement units (IMUs) combined with GPS to achieve data pose recovery and estimation for autonomous navigation. However, in many environments of interest, one of the most critical issues for autonomous navigation is the unavailability of GPS signal measurements capable of providing a global localization. Moreover, not only must the system function without GPS, but it might also need to detect the presence of a multitude of obstacles including buildings, vehicles, pedestrians, trees, and power lines by generating terrain maps and creating a flight plan.

To generate a non-GPS dependent estimation, several sets of vision-based algorithms have been designed. However, the estimation of certain states tends to drift with time, given that no absolute measurement has been provided, due to the integration of errors with respect to time. Therefore, the success of such algorithms depends on how accurately the system generates the estimation before the drift is significant or a GPS measurement is recovered. Several vision-based techniques have been proposed in the past by using information about the light intensity, terrain elevation, landmark geometry, or target locations. However, such algorithms rely on previous knowledge about the environment where the vehicle navigation is going to occur.

For cases where no prior information is available, one can use the inter-frame relationship of random feature points to reconstruct the pose of the vehicle based on projection models from which the attitude and translation are extracted. A common projection model used for this estimation is called the epipolar geometry, from which the homography relationship is also derived (Ma et al., 2004; Brockers et al., 2011; Mondragón et al., 2010). However, the main problem with these algorithms is that the estimates tend to drift away from the real solution due to their relativistic nature (i.e., epipolar geometry and homography provide information about the frame-to-frame camera motion). A common approach used to address this problem is the broad class of simultaneous localization and mapping (SLAM) algorithms where registration methods, such as constructing image mosaics, are used in conjunction with visual odometry and inertial navigation systems (INS). Therefore, a map based on the estimation of the tracked features can be generated to relate this map with the attitude and the pose of the vehicle (Indelman et al., 2010; Caballero et al., 2007). However, traditional SLAM algorithms are limited by their computational cost, which could limit their viability for space applications.

1.2 Purpose

With the objective of addressing the need for a reliable navigation system that does not depend on GPS measurements, this thesis focuses on the development of vision-based state estimation algorithms based on the planar homography relationship. With this approach, image processing algorithms are first applied to track planar features in image frames collected by an onboard monocular camera. The homography relationship is then applied to provide vision-based pose reconstruction measurements, which correspond to the frame-to-frame rotation matrix, the unit vector of frame-to-frame translation, and the unit normal vector to the feature point plane.

An extended Kalman filter (EKF) implementation is developed as a sensor fusion algorithm that uses accelerometer and rate gyro data from an onboard inertial measurement unit (IMU) in the state propagation step and homography-based measurements, barometer, and magnetometer data in the measurement update step. Additionally, using the estimated states, a homography relationship is reconstructed.

Finally, a hybrid-based approach is proposed using the homography measurement and known tracked landmarks as an absolute measurement for mitigating drift accumulation. The implementation of these algorithms aims to accomplish the main objectives of this thesis:

1. *Design a vision based nonlinear state estimator, based on tracked features and planar homography, for the autonomous navigation of a flight vehicle in unknown environments.*

2. *Investigate the limitations of the proposed algorithm and possible solutions for the mitigation of such limitations without significantly adding to the computational cost of the algorithm.*

3. *Demonstrate autonomous navigation under GPS denied environments with the usage of inertial and vision measurements.*

While it is true that a successful implementation of a vision-based approach for navigation requires a consideration of hardware design and optimization, this thesis is focused on the algorithmic side of the problem statement. Therefore, the research presented is oriented towards the study of the sensor fusion, but not on the platform-dependent control of the vehicle. The design of the control algorithm strategies can be found in different implementations such as in (Garcia Herrera, 2017) or (Perez Rocha et al., 2016).

2. Literature Review

Aerial multi-rotor vehicles are vehicles for which propulsion is generated by a set of propellers configured in different symmetries. One of the most typical configurations is the X4 configuration, commonly known as quadrotors due to the number of pitch-rotors that are mounted on the vehicle. This type of vehicle is omnidirectional and has almost no constraints in its motion. The quadrotor was first introduced in 1922 by Dr. George de Botherzat and Ivan Jerome. However, its mechanical complexity, unresponsiveness, and inherent instability made the vehicle a hard to control platform where the pilot workload was significant (Hoffmann et al., 2004). Therefore, the complete development and further popularization of quadrotors did not occur until the last few decades with the introduction of micro electro mechanical systems (MEMS), gyroscopic stabilization, and digital microcontrollers, that enabled for the miniaturization of the platforms and the development of autonomy for the vehicles (Garcia Herrera, 2017).

The autonomous navigation of micro aerial vehicles (MAV) problem emerged as an important research topic in recent years. MAVs can provide access to environments where no humans can enter, thus minimizing or voiding human risk because MAVs have considerable 3D maneuverability and portability. The special capabilities of autonomous vehicles provide an ideal solution for search and rescue, asteroid mining, or simple reconnaissance. However, the success of any autonomous vehicle mission depends on the ability to navigate the scene by sensing the environment using active or passive sensors.

Although a consistent definition for navigation has not been agreed upon, some pundits have defined navigation as a system composed of one or several methods to provide position and velocity of an object with respect to a given reference frame or point. However, it is important to define the difference between positioning and localization. Positioning refers to the quantitative determination of the position of a vehicle with respect to a coordinate system or a map, whereas localization is often referred to as a method for the qualitative determination of position, thus constraining the position of the vehicle to an area (Groves, 2008). Different methods for localization and positioning are usually used interchangeably depending on the application. This chapter presents an overview of some common algorithms used for the autonomous navigation and attitude estimation of MAVs using either inertial, remote sensing, or vision-based measurements.

2.1 Dead Reckoning (DR)

Dead Reckoning (DR) refers to the process of computing the current position of an object based on the measurement of the change of position or the integration of a given or estimated velocity with respect to time. The DR process can be accomplished by different means. For example, for the estimation of the attitude, a separate solution is usually used depending on the movement constraints of the vehicle. For 2D navigation, a magnetometer heading is usually sufficient. However, when the vehicle undergoes a 3D translation, it is required to measure the three axis angles using gyroscopes. For pedestrian dead reckoning, the process is accomplished by counting paces using a pedometer or accelerometers. For vehicles with 3-DoF, the DR process can be accomplished by the usage of an odometer that uses measurements of the distance travelled by the vehicle using the count of rotations of the wheels. The successful implementation of a DR solution onto a system depends on the accuracy and precision of the sensors used since it is an algorithm that depends on the size of the time step that is used for the integration (Groves, 2008). Therefore, the main challenge in DR is the accumulation of possible drift due to measurement noise that could lead to errors of 10% or more of the distance travelled, depending on the sensors used.DR is a process where the error grows with time because DR navigation usually involves the integration of several inertial measurements. The application of Bayesian filtering methods are sometimes used for drift reduction (Diamant & Jin, 2014). A description of the DR accumulation of error process is shown in Figure 2.1.

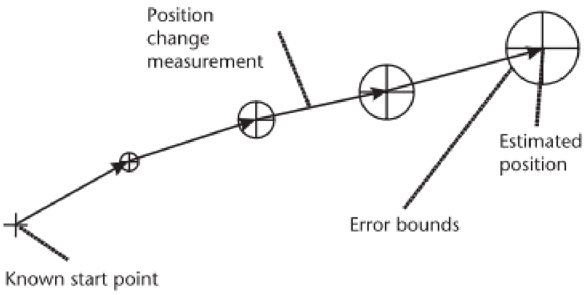


Figure 2.1: Dead Reckoning Drift Accumulation Process (Groves, 2008)

2.1.1 Inertial Navigation System (INS)

An inertial navigation system (INS) is also commonly referred to as an inertial measurement unit (IMU), and it is a complete DR system that is usually composed of three mutually orthogonal accelerometers, three-axis gyroscopes, and sometimes a set of three mutually orthogonal magnetometers as shown in Figure 2.2. The measurements are usually processed by a microcontroller that generates a DR solution for position, orientation, and velocity (Groves, 2008).

The inertial solution for position and velocity is computed using Newtonian mechanics to calculate the motion of the vehicle by measuring the external forces on the vehicle that cause an acceleration and excluding the effects of gravity. For attitude determination, the gyroscopes are used as a way of sensing the rotational velocity of the vehicle given by the gyroscopic effect. The rotational rates are then used in conjunction with the direction cosine matrix (DCM) to relate the vehicle body reference frame to the inertial reference coordinate system using numerical integration. This process is also called a strapdown navigation system (Titterton & Weston, 2004).

2.2 Signal-Based Navigation

Signal-based navigation is comprised of a set of different passive sensors that are used in conjunction with onboard systems. Such systems are usually based on the triangulation of the signal being sent and received by a set of transmitters and an onboard receiver

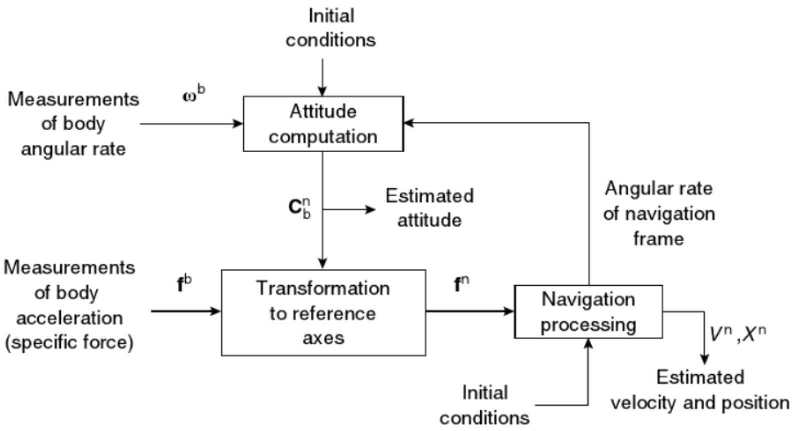

Initial conditions
Measurements of body angular rate
ω^b
Attitude computation
C_b^n
Estimated attitude
Angular rate of navigation frame
Measurements of body acceleration (specific force)
f^b
Transformation to reference axes
f^n
Navigation processing
V^n, X^n
Estimated velocity and position
Initial conditions

Figure 2.2: Strapdown Inertial Navigation System Diagram (Titterton & Weston, 2004)

respectively. The two main categories within signal-based navigation are radio navigation and satellite navigation, usually known as the Global Positioning System (GPS).

2.2.1 Radio Navigation

Signal-based navigation started in the form of radio navigation in the 1920s with the transmission of low- and medium-frequency signals with marker beacons running at 75MHz that allowed researchers to delineate airways (Groves, 2013). The main purpose of radio navigation is the determination of the position of an object. Nowadays the radio navigation is composed of more complex techniques that involve wireless local area network (WLAN) or ultrawideband communications. Five of the most common basic radio navigation techniques still in use include: radio direction finding, bearing, passive ranging, hyperbolic ranging, and Doppler signal positioning (Groves, 2008).

Radio Direction

Radio direction systems work by the determination of the direction by tuning in a broadcasted radio station with a rotatable directional antenna linked to a degree indicator to obtain a bearing (Groves, 2008). Additionally, a second measurement of the same type can be used as means of triangulation for position determination as shown in Figure 2.3. This is a typical system found in commercial aircraft and maritime vessels because the broadcasting signal generators are usually found near airports or harbors. The main difference between radio direction and bearing is the fact that the reference station emits a signal that varies with the direction of transmission, allowing the vehicle to compute the elevation to the reference station without the usage of a directional antenna attached to the vehicle (Groves, 2008).

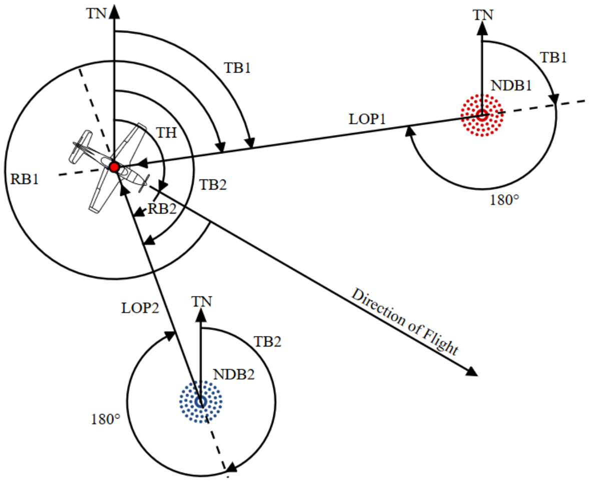


Figure 2.3: Radio Direction Triangulation (Indutiveload, 2008)

Passive Ranging

For passive ranging systems, the computation is done by measuring the time of arrival of the signal being broadcasted from the reference station. A range from the transmitter can be deduced by the usage of a timing signal. The output of such systems is 2D tracking information that can contain bearing and elevation rate (de Visser et al., 2006). Therefore, it is necessary that the clocks of the receiver and transmitters are synchronized (Groves, 2008).

Hyperbolic Ranging

The basis for hyperbolic ranging systems is the estimation of the location of a transmitting source using the intersection of the hyperboloids that describe the difference in distance measurements of at least three reference stations. The distance difference is computed by measuring the time of arrival difference between the three or more signals sent (Polturi, 2007). A graphical description of the process is shown in Figure 2.4.

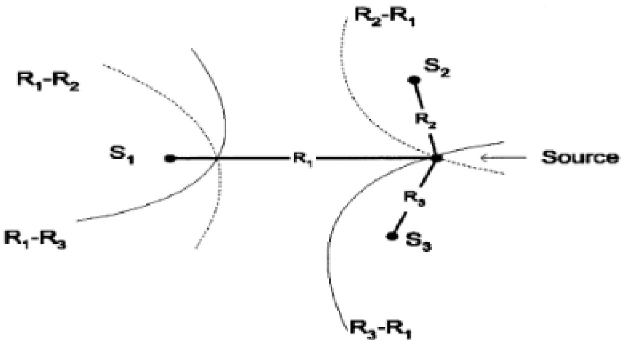


Figure 2.4: Hyperbolic Ranging Position Localization (Polturi, 2007)

Doppler Positioning

The most used radio navigation method is Doppler-based positioning. This method uses a transmitter that moves along a predetermined trajectory while a receiver is measuring the Doppler shift of the signal being received (Groves, 2008). When the transmitter approaches the receiver, the transmitted signal bounces with an increment in the frequency. This bounce can be modelled as

$$\omega' = \left(1 - \frac{v_{rel}\cos(\theta)}{v_{prop}}\right)\omega \tag{2.1}$$

where ω' represents the received frequency, ω the transmitted frequency, v_{rel} is the relative velocity, v_{prop} decribes the speed propagation of the wave, and θ is the angle between the relative velocity and the wave propagation direction. Using the frequency shift representation, it is possible to compute the translation of an object as shown in Figure 2.5 (Lehtinen, 2001).

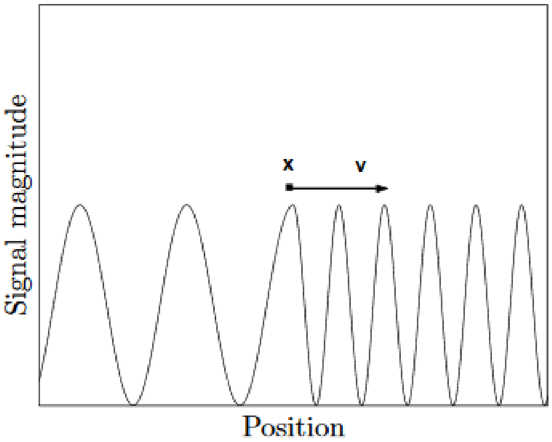
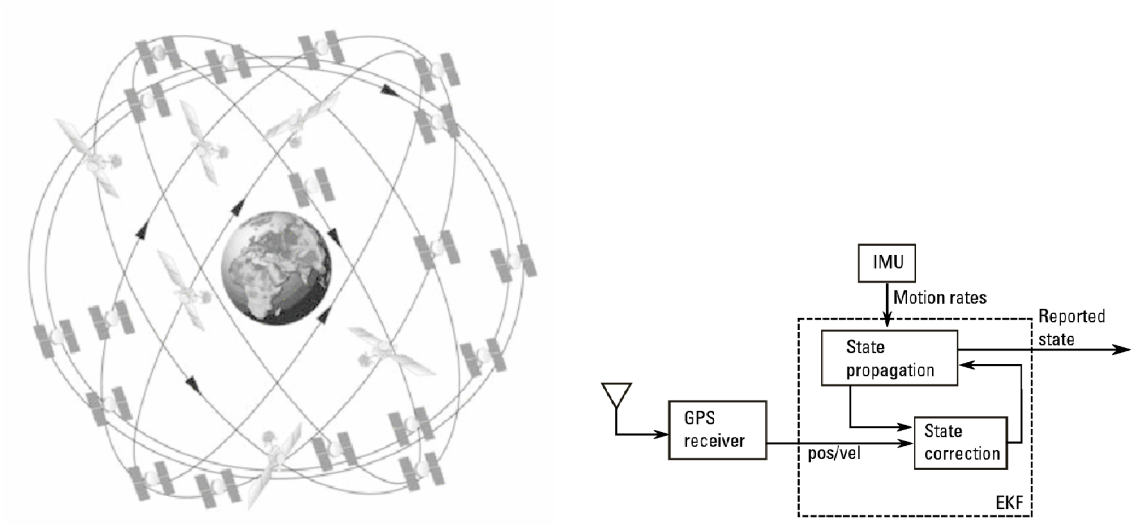


Figure 2.5: Frequency shift of a moving object (Lehtinen, 2001)

The main problem with the usage of radio-based positioning at low to medium frequencies is the coverage of large areas since these frequencies can only transmit signals within line-of-sight because of the effect that atmospheric conditions produce. (Titterton & Weston, 2004).

2.2.2 Satellite Navigation

Global navigation satellite systems are a constellation of typically 20 to 30 satellites that fly in three to six different orbital planes close to medium earth orbits (MEO) as shown in Figure 2.6(a) (Bevly & Cobb., 2010). The main objective of these systems is to generate a differential signal of at least four satellites that can be acquired using a receiver at any location. The system generates a three-dimensional position fix using the satellite broadcasted microwave signals and calibrates its clock bias using passive ranging methods. This method can provide accuracy of a few meters with the ability to increase its precision and accuracy by using carrier-phase positioning techniques or differential GPS (Groves, 2008). Additionally, GNSS measurements can be fused with IMU measurements using Kalman Filters (KF) or other Gaussian filtering techniques to generate a more precise solution as shown in Figure 2.6(b). However, the main problem with this type of systems is the sensitivity to interference and occlusion generated by structures.



(a) GNSS Satellite Constellation (Bevly & Cobb., 2010)

(b) GNSS with INS integration diagram (Bevly & Cobb., 2010)

Figure 2.6: GNSS Positioning Description

2.3 Vision-Based Navigation (VBN)

Vision-based navigation (VBN) techniques use optical sensors to extract information about the environment, so they can be post-processed with computer vision algorithms to provide a positioning or localization solution. These systems have become popular lately, especially during the last decade, due to the economic viability of its hardware and software. The main advantage of vision-based navigation techniques is the versatility that these methods provide. Therefore, vision-based techniques are not only used for localization, but also for mapping, collision avoidance, and path planning. However, the main disadvantages of these algorithms is the low reliability and low updating rate when compared with INS systems. Therefore, VBN systems are commonly used in conjunction with other navigation systems, such as GNSS and INS, to improve accuracy (Sabatini R, 2013). These algorithms can be classified into two categories, algorithms with a priori information, also called model-based approaches, and those without previous information, also named appearance-based approaches (Zhao et al., 2013; Sabatini R, 2013).

2.3.1 Algorithms with Model-Based Approach (MBA)

VBN systems with model-based approach are systems that use previously stored information as a source for orientation or visual scaling using either 2D or 3D correspondences. MBA methods can use previous information about the environment other than knowing the geometry of the objects being tracked, but also information such as light intensity, terrain elevation, target positioning, or even reference maps that could be used to generate an absolute estimate for localization. This approach is commonly referred to as environmental feature matching. One of the main characteristics of this class of algorithms is that there is usually some preprocessing involved. Environmental feature matching can fail to provide a navigation solution when the information about features or the environment fail to match the database. Some MBA vision-based approaches are listed below:

Vision Bearing

This system is based on proprioceptive geometry or location data of an artificial landmark. Knowing the geometry of the landmark makes the camera the equivalent of a bearing sensor (Caballero et al., 2009). The pose is usually recovered by solving the well-known structure from motion problem in which the attitude and position of the vehicle are recovered using camera projection models that include the camera intrinsic and extrinsic parameters. A straightforward approach of these systems is to identify detectors based on the appearance, color, and shapes of an object captured by the camera and then use a comparison with templates included in an internal library (Lin et al., 2009). An example of this method was developed by DeAngelo & Horn (2016) where a landmark detection system was developed using information from previously taken aerial images. In this example, the computer was trained to recognize visible features of the aerial images, such as roads. These detectors were used in conjunction with close aircraft region location estimation that was used to recognize the landmark detectors for a precise localization computation of the flying vehicle. A general system diagram for a vision bearing navigation approach is shown in Figure 2.7 where the vision bearing system is used as a position fixing method.

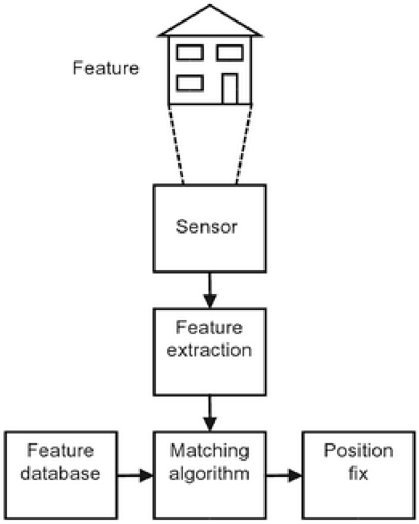


Figure 2.7: Vision bearing system (Groves, 2013)

Map Matching

The idea of map matching is to use a map database in conjunction with a feature matching algorithm given a set of information that can be in the form of prelocalization. Therefore, this algorithm is commonly used with georeferenced images and INS solutions. Map matching compares the input position solution from the rest of the navigation system with roads, houses, or scenery in the environment that can be used as a reference localization with respect to the database map (Zhao et al., 2012). A visual representation of feature extraction, templating, and matching is shown in Figure 2.8. In order to generate these correspondences, algorithms such as Canny edge detectors or Progressive Probabilistic Hough Transforms are used to extract the features of the environment (Kuemmerle et al., 2011). In (Dogruer et al., 2008) satellite images from Google Earth were segmented into separate regions to distinguish common urban features such as roads and buildings. Then a Monte-Carlo Localization was used in conjunction with the segmented map images to generate an estimate of position.

For digital road matching, the comparison is done by means of coordinate fixing from GNSS together with a road link identification. In this system, the map database is represented in the form of grids and nodes. The main objective is to pinpoint the location of the vehicle to a node or connection based on fuzzy logic, probabilistic theory, or Bayesian inference (Groves, 2013). However, this system is mainly suited for ground vehicles only in urban or known environments.

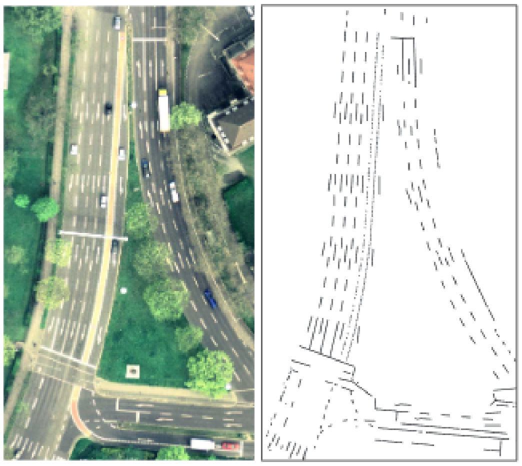


Figure 2.8: Map feature comparison (Pink, 2008)

Terrain-Referenced Navigation (TRN)

Terrain-Referenced Navigation (TRN) uses terrain pattern matching in comparison with a terrain database. The comparison is done using terrain height measurements as shown in Figure 2.9. Therefore, this system is also referred to as terrain-contour matching. The methods for obtaining the position from the measurements can be classified as sequential and batch processing. For sequential processing, an EKF is usually used for error estimation with measurements coming from radar altimeters. For the case of Batch processing, the terrain contour is generated using 5 to 16 terrain height measurements so a location signature in the form of a footprint can be generated, followed by a correlation process using a probability distribution that relates geolocalized images with the position of the vehicle.

In 2012 (D. Lee et al., 2012) used a digital terrain database with a monocular camera. The UAV position estimation was computed using a radar altimeter to recover the depth perception lost by the monocular camera. The measurements were integrated using a point-mass filter. Additionally, the INS was replaced by the usage of vision-based odometry (homography decomposition).

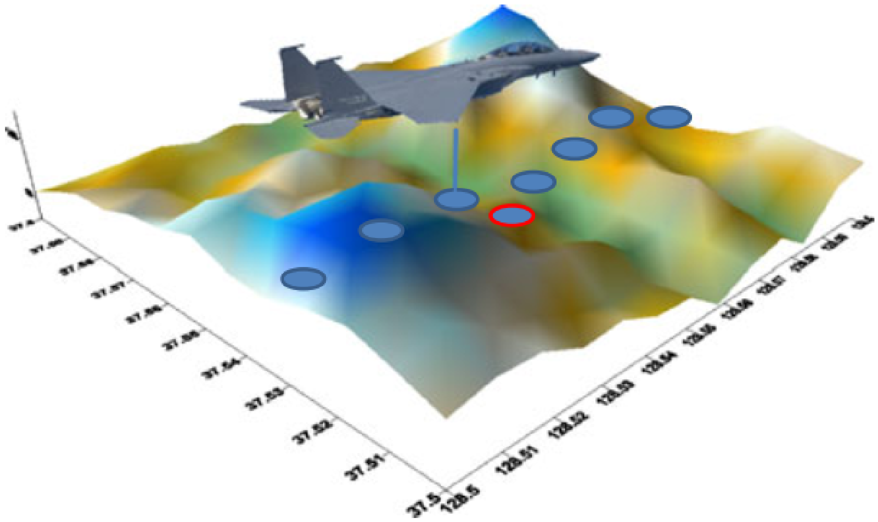


Figure 2.9: Vision-based Terrain-Referenced Navigation (D. Lee et al., 2012)

2.3.2 Algorithms with Appearance-Based Approach (ABA)

Appearance-based vision navigation systems are designed to recollect as much information as possible from the image so an estimation of the followed path and attitude can be computed. This is accomplished by relating previous frames with the current one. Therefore, it can be inferred that the environment is represented in the form of key images taken at various locations using visual sensors (Sabatini R, 2013). The information recollected from each frame is stored in a database as a series of small- feature descriptors that can be used for matching in subsequent images. The main advantage of ABA over MBA is that ABA usually does not need any kind of preprocessing. Consequently, this class of algorithms can be used in virtually any unknown environment. However, the main problem with ABA approaches is that they are usually computationally expensive and they carry a larger uncertainty than with MBA approaches. A few approaches for ABA are listed as follow:

Vision-Based Odometry (VO)

Visual odometry is a form of dead reckoning that indirectly measures the translation and orientation of the vehicle by relating camera images. The translation and orientation measurements are done by using projection models between vectorial subspaces represented as camera frames at two instances of time (homography matrix) or camera projection model for the current time (camera fundamental matrix). The geometry used for the generation of the projection models is referred to as the epipolar geometry or two-view geometry. The epipolar geometry is the intrinsic projective geometry between two views and is independent of scene structure, thus it can be imagined as the projection of a point in 3D space onto two sequential images as represented in Figure 2.10. The epipolar geometry just depends on the internal parameters of the camera and relative pose. However, the projective epipolar geometry can be reconstructed using either the 4-points or 8-points algorithms (Choi et al., 2015). The projection model between two frames can be represented as

$$\begin{pmatrix} {x^{c}_{cf}}^{(+)} \\ {y^{c}_{cf}}^{(+)} \\ 1 \end{pmatrix} = H \begin{pmatrix} {x^{c}_{cf}}^{(-)} \\ {y^{c}_{cf}}^{(-)} \\ 1 \end{pmatrix} \tag{2.2}$$

where $\left(x_{cf}^{c}{}^{(+)}, y_{cf}^{c}{}^{(+)}, 1\right)$ homogeneous coordinates represent the position of a 3D projected point onto the image frame in camera coordinates. The subscripts $(+)$ and $(-)$ denotes the previous and current frames respectively. The H matrix is the projective matrix that maps the points from the previous to the current frame. This is called the homography matrix and encodes information about interframe relative rotation and translation that can be used to generate a DR solution for attitude and position estimation (Groves, 2013). The accuracy of visual odometry with monocular vision is bounded to approximately 2% of the distance traveled by the vehicle (Hide et al., 2010).

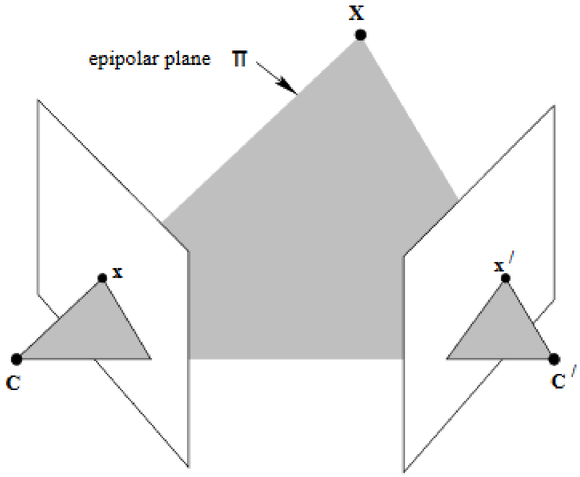
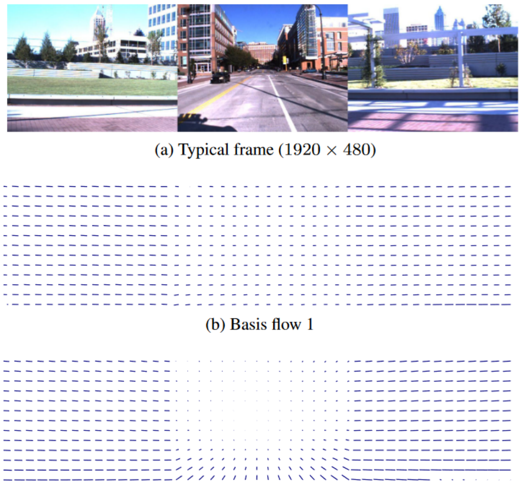


Figure 2.10: Two-View Projective Epipolar Geometry Representation (Ma et al., 2004)

Optical Flow

Optical flow egomotion can be also classified as a visual odometry algorithm. This algorithm computes the apparent motion of objects, surfaces, or edges between two consecutive frames caused by the relative displacement between an object and the camera. It is usually represented as a vector field that represents the differential displacement of the tracked object as a function of time. Therefore, one of the most popular computer vision algorithms for the computation of optical flow is Lucas-Kanade method due to its differential behavior (Bradski, 2000). However, the vector field representation can also be energy-based or phase-based. An example of differential and energy field representation is shown in Figure 2.11.



(a) Typical frame (1920 × 480)

(b) Basis flow 1

(c) Basis flow 2

Figure 2.11: Optical Flow Differential and Energy-Based Potential Field Representation (Roberts et al., 2009)

Optical flow methods can be computed by analyzing the pixel intensity given as a function of the location (x,y) in the pixel frame and time t in the following manner

$$I(x,y,t) = I(x+dx, y+dy, t+dt) \tag{2.3}$$

from which the potential field differentiation can be computed by a Taylor series approximation as follow

$$f_x u + f_y v + ft = 0 \tag{2.4}$$

where fx and fy are image pixel gradients. Consequently, fx and fy represent gradients with respect to time as represented below

$$\begin{aligned} f_x = \frac{\partial f}{\partial x} \quad f_y = \frac{\partial f}{\partial y} \\ u = \frac{dx}{dt} \quad v = \frac{dy}{dt} \end{aligned} \tag{2.5}$$

(Bradski, 2000)

However, in order to generate a suitable measurement, this algorithm depends on a series of assumptions that ensures that the optical flow principle can be formulated. This algorithm assumes that the environment does not change, or changes smoothly. Additionally, the displacement between frames of the tracked object is small compared to the size of the image (Garcia Herrera, 2017).

The gradients associated with the image optical flow can be used to relate the angular velocity of the vehicle and the translational motion with the optical flow. Despite the direct measurement of the vehicle's translation and rotation, the visual measurement solution tends to drift with time due to its relativistic nature. Therefore, it is necessary to fuse the optical flow solution with inertial measurements to recover the scale ambiguity, caused by the loss of depth perception, and to correct the drift.

Simultaneous Localization and Mapping (SLAM)

As its names suggest, SLAM is regarded as an estimation problem in which the location and pose of the vehicle are estimated using all previous sensor readings and all previous sensor actions. In the SLAM framework, it is often assumed that the localization process obeys a Markov approximation assumption. This means that the world is assumed to be static, noise is independent, and that no approximation errors during modeling are made. Additionally, the mapping part of the algorithm also includes an estimation component. The map estimation consists of n landmarks that are independent of GPS measurements, making this algorithm a suitable solution for position and heading drift. In addition, this algorithm can be used in known or unknown environments. For known environments, the pose uncertainty can be constrained so the observation of landmarks can be used for a bounded precise pose estimation. For unknown environments, the uncertainty can increase arbitrarily due to the summation of errors in the odometry. To tackle this problem, SLAM algorithms detect and store a position where the vehicle has previously been before so a landmark matching can be performed. The main objective is to bound the error to certain limits (Nütcher, 2001).

In order to generate a translational and rotational measurement, SLAM algorithms are commonly used with visual odometry solutions using either monocular cameras, stereo vision systems, or light detection and ranging (LiDAR) sensors. The advantage of using stereo vision systems or LiDAR sensors over monocular vision is that depth perception is provided. Therefore, it provides a complete solution to the SLAM problem. However, the complexity of the algorithm increases on the computational side.

The most common approach for SLAM uses an extended Kalman filter (EKF) since the filter is capable of integrating all landmark position measurements in the form of a covariance matrix that can be related to the states of the vehicle. The covariance relationship used by the SLAM EKF is shown in Figure 2.12. For linear measurement models, the EKF yields a maximum likelihood estimation, whereas for nonlinear models, a point linearization is computed. However, when the vehicle moves through unmapped areas, the uncertainty of the point around which the model is linearized can be too high. Therefore it is important to generate closed loops when generating a SLAM solution (Frese & Hirzinger, 2001).

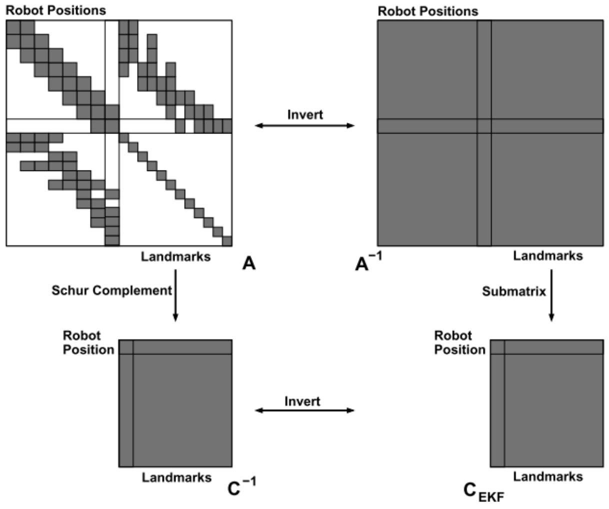


Figure 2.12: Relationship Between Process Information Matrix and Covariance Matrix in SLAM EKF (Frese & Hirzinger, 2001)

3. Projective Geometry

3.1 Geometric Primitives

A point $X = (X,Y,Z) \in R^3$ in the 3D Euclidean space, representing the local coordinates of a feature in the scene, can be represented as a projection onto the camera plane as shown in Figure 3.1. Specifically, this projection is a 2D representation in the camera coordinate frame. This camera projection is usually denoted as $x = (x,y) \in P^2$, where P^2 represents 2D space. This point representation can be expressed in terms of homogeneous coordinates where the overall scaling is irrelevant. In this manner, any pair of values representing a point in homogeneous coordinates can be represented as $x = (kx,ky,k)$, where k is any non-zero value that contains scaling information about the points (Szeliski, 2010). Thus the vector x can be expressed as:

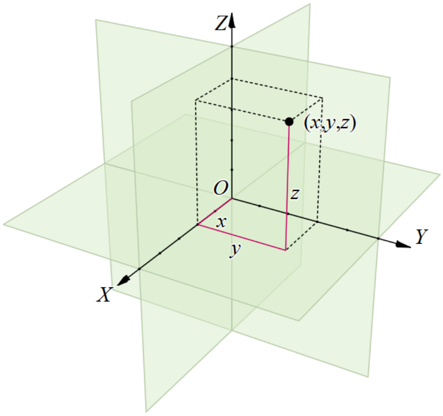


Figure 3.1: 3D Point Projection Representation (Stolfi, 2009)

$$\bar{x}_h = k(x, y, 1) = k\bar{x} \tag{3.1}$$

Similar to the 2D case, points that belong to the R^3 space can also be represented in homogeneous coordinates where a fourth term is added to represent the scale of the point. Thus, a vector $X \in R^3$ can be expressed in homogeneous coordinates as:

$$X_h = k(X, Y, 1) = kX \tag{3.2}$$

In order to represent the vector X on the 2D projective space, it is necessary to map the space R^3 to P^2. This is done using the projective matrix, which in the computer vision world is usually modeled as the pinhole model. Using the pinhole camera model, a point in 3D homogeneous coordinates can be projected to the image plane. The projective pinhole camera model can be expressed as:

$$\bar{x}_h = K_{intr} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} X_h \tag{3.3}$$

where R and T represent the rotation and translation of the camera in the world coordinate frame respectively. Additionally, K_{intr} represents the intrinsic camera matrix that contains information about the image plane, scaling factors, radial distortion, and pixel center (Zhang, 2002).

3.2 Homography Relationship

Consider a 2D plane in the 3D space and a camera looking at a set of feature points P at two different frames. The projections of P onto frames 1 and 2 are represented as $X_i \in R^2$ and $X_i' \in R^2$ respectively, as shown in Figure 3.2. It is possible to relate the projection X_i to X_i' using a proportional relation as shown by Eq. 3.4.

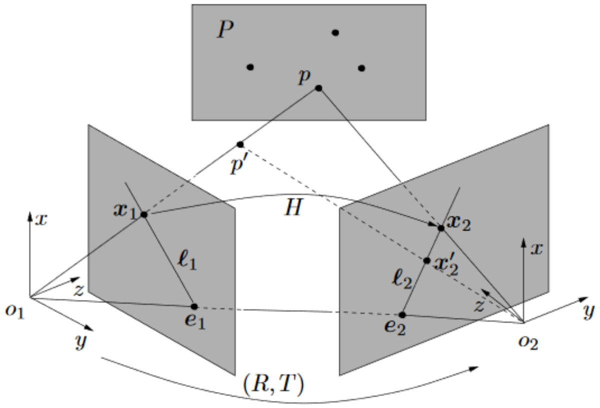


Figure 3.2: Planar Homography Relationship (Ma et al., 2004)

$$X_i' =\sim HX_i \tag{3.4}$$

From Eq. 3.4 it can be deduced that a mapping from R^2 is projective if and only if there exists a non-singular matrix H of size 3×3 such that for any point X in the plane R^2 there exists a mapped correspondence of the form HX_i. It should be noticed from Eq. 3.4 that H can be multiplied by any scalar without altering the projective transformation. Thus it can be inferred that the projective homography matrix only has 8-DoF despite having 9 elements, or in other words, the homorgaphy matrix has a scale ambiguiy associated with it.

This means that there exist 8 unique unknowns parameters for which the system needs to be solved (Dubrofsky, 2009; Ma et al., 2004; Hartley & Zisserman, 2003).

3.2.1 Direct Linear Transformation (DLT)

It can be demonstrated that, in order to compute the homography matrix, a set of at least 4 coplanar tracked feature points is required (Ma et al., 2004). Given at least 4 coplanar points, the homography relationship can be used to project the set of points from frame 1 to frame 2. The homography relationship contains information about the frame-to-frame camera rotation, the unit vector of camera translation, and the unit vector normal to the feature point plane. Given a static feature point P that has been tracked in two image frames, the 3×3 homography matrix H relates the measurement of P in frame 1 to its measured location in frame 2 as follows:

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \tag{3.5}$$

where $(x_n, y_n, 1)$ is the homogeneous representation of the coordinates of the point P in the frames 1 and 2 respectively.

If Eq. 3.5 is written for all the tracked planar feature points, it can be written in the form:

$$A_i h = 0 \tag{3.6}$$

where $h \in R^{9\times 1}$ is the vector form of the stacked columns of the homography matrix $H \in R^{3\times 3}$ and A_i is a matrix composed of feature point measurements in frames 1 and 2 defined in Eq. 3.7. Eq. 3.6 can then be solved for the homography matrix parameters. This is called the direct linear transformation (DLT) (Ma et al., 2004).

$$A_i = \begin{pmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_2x_1 & x_2y_1 & x_2 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & y_2x_1 & y_2y_1 & y_2 \end{pmatrix} \tag{3.7}$$

At least 4 correspondences are required to compute the null space as the solution for the 8-DoF of the matrix H. However, when more than 4 correspondences are used, the system becomes over-constrained. In order to solve for the homography parameters, it is necessary to compute the singular value decomposition (SVD) of A so the null space can be computed. However, the main problem with SVD in DLT is that the algorithm is dependent on the origin and scale of the image coordinate system, making the algorithm numerically unstable. In order to address this problem, it is necessary to generate a similarity transform T that maps the set of points $X_i \in R^2$ to a new set $\hat{X}_i$ with centroid at the origin and average Euclidean distance of $\sqrt{2}$ in frames 1 and 2 respectively. After the computation of the similarity transform T, the SVD can be used to compute the homography matrix $\widehat{H}$ (Hartley & Zisserman, 2003). It is then necessary to map the homography back using the similarity transform as shown below

$$H = (T)^{-1}\widehat{H}T \tag{3.8}$$

In order to eliminate the scale ambiguity, it is recommended to normalize the homography matrix H dividing all of the elements of the matrix by its last element H_{33}, which corresponds to the smallest singular value of A. This way a constraint on the vector h is enforced so that the algebraic distance is minimized (Dubrofsky, 2009).

3.2.2 Nonlinear Homography

When the measurement points used for the computation of the homography matrix contain noise, the problem for the homography computation turns into an optimization problem that minimizes the reprojection error using the distance error (d) as the cost function. This error distance is used to generate the Maximum Likelihood estimation (MLE) of $\hat{H}$ given $n > 4$ image point correspondences $\{X_i \leftrightarrow X_i'\}$. However, the cost function can take different forms as discussed below.

Geometric Distance

The idea of the geometric distance is to measure the Euclidean distance (d) between the reprojection of the points X_i in frame 1 using the estimated homography matrix $\hat{H}$ and the correspondence X_i' as follows

$$\sum_i d(X_i', \hat{H}X_i)^2 \tag{3.9}$$

Similarly, the geometric distance can be computed in both directions. This is called *symmetric transfer error* and it measures the reprojection errors forward and backward. This is calculated as:

$$\sum_i \left\{ d\left(X_i', \hat{H}X_i\right)^2 + d\left(X_i, \hat{H}^{-1}X_i'\right)^2 \right\} \tag{3.10}$$

In order to generate an iterative solution for the MLE, it is necessary to compute an initial estimate, usually based on the DLT, that can ensure convergence. However, the stability of the optimization process is not ensured (Dubrofsky, 2009).

Reprojection Error

The reprojection error cost function is used as a measure of error distance between the projected feature correspondence using the estimated $\hat{H}$ matrix. The error distance is computed for each of the features. This method aims to generate an optimal estimation of $\hat{H}$ and the correspondences $\{\hat{X}_i \leftrightarrow \hat{X}'_i\}$ such that Eq. 3.11 is minimized (Dubrofsky, 2009) (Hartley & Zisserman, 2003).

$$\sum_i \{d(X_i, \hat{X}_i)^2 + d(X'_i, \hat{X}'_i)^2\} \tag{3.11}$$

given that

$$\hat{X}'_i = \hat{H}\hat{X}_i$$

The main problem with this approach is that the computational cost of the algorithm increases due to the calculation of the geometric distance in two directions.

Sampson Approximation

The main objective of the Sampson approximation is to generate an optimal estimation of the reprojection error using only 9 parameters corresponding to each of the entries of the H matrix. This is accomplished by representing the point correspondence $(X_i \leftrightarrow X'_i) \in R^2$ as a 4D point $(X_i \in R^4)$. This defines the error cost function a representation of an algebraic curve V_H that passes through the points X_i. The function to minimize then becomes

$$\sum_i ||X_i - \hat{X}_i||^2 \tag{3.12}$$

where the vector $\hat{X}_i$ represents the estimated closest point to X_i on V_H using a Taylor series expansion (Dubrofsky, 2009). In practice, this approximation gives precise results given that the errors are small compared to the measurements (Hartley & Zisserman, 2003).

3.2.3 Robust Homography

The computation of the homography matrix H so far has been assumed that the only error present in the computation comes from errors in the measurement such as pixel noise in the cameras, which is assumed to follow a Gaussian distribution. However, the feature matching algorithms almost never yield a set of matching correspondences with 100% reliability. That means that there is usually a set of outliers that have been computed as a matching correspondence and are outside the Gaussian distribution. Such mismatches are capable of generating a disturbed homography solution with high values of reprojection error. Therefore, the goal is to generate a robust homography by determining the set of inliers using the presented correspondences and then estimating the optimal homography using the computed inliers. The process is done by using an algorithm called Random Sample Consensus (RANSAC).

The idea of RANSAC is to uniformly select a random subset of the data sample and estimate the model parameters. Then the algorithm determines the samples that are constrained within an error tolerance (inliers). If the number of inliers is high enough (minimum four inliers for homography), the algorithm proceeds to generate a final model of the consensus. If after a number of iterations, the minimum number of inliers has not been found, the algorithm returns the model which has the smallest average error among the iterations. For the specific case of homography computation, the tolerance error is computed by using the reprojection error model from Eq. 3.11. Additionally, a Least Mean Square (LMS) algorithm is commonly used in conjunction with RANSAC as described by Hartley & Zisserman (2003), since it requires no expected initialization error. Thus the algorithm to compute a robust homography is as follows:

Algorithm 3.1: Homography robust estimation pseudocode (Brown, n.d.)

```
Input: $n > 4$ point correspondences $\{X_i \leftrightarrow X_i'\} \in R^2$
Output: $\hat{H}$ between Frame 1 and 2
Initialization: Compute an initial estimate of $\hat{H}$ using DLT ;
$nBest = 0$;
for $i = 0;\ i \leq MaxNumIter;\ i++$ do
    Select random subset of data points $S_i$;
    Compute nonlinear homography $\hat{H}_i$;
    Compute inliers $(S_{inliers})$ given $\hat{H}_i$ using reprojection error;
    if $size(S_{inliers}) > nBest$ then
        $\hat{H} = \hat{H}_i$;
        $nBest = size(S_{inliers})$;
```

3.2.4 Homography Decomposition

Given a set of 3D points $P \in R^3$ projected on two subsequent images $\{X_1, X_2\}$ one can compute the coordinate transformation of $X_i \to X_i'$ by rotating and translating the points X_i to the pose of X_i' by

$$X_i' = R_1^2 X_i + T_{2\to 1}^2$$

where X_i and X_i' represent the the coordinates of P relative to frames 1 and 2 respectively, R_1^2 is the rotation matrix from camera frame 1 to camera frame 2, and $T_{2\to 1}^2$ is the translation vector of the camera from frame 2 to frame 1, expressed in the camera 2 frame. However, if the the set of points P are in the same plane, one can deduce the following equation:

$$X_i' = \left(R_1^2 + \frac{1}{d} T_{2\to 1}^2 N^T\right) X_i$$

where $\frac{N}{d}$ represents the unit vector normal to the feature point plane in camera frame 1. Additionally, it can be inferred that $R_1^2 + \frac{1}{d} T_{2\to 1}^2 N^T$ is just a constant giving the same result as from Eq. 3.6. Therefore, the homography matrix can be defined as follows:

$$H = R_1^2 + \frac{1}{d} T_{2\to 1}^2 N^T \tag{3.13}$$

From Eq. 3.13 it is possible to observe that the homography matrix contains information about the translation and rotation of the camera between camera frames 1 and 2 respectively, as shown in Figure 3.2. However, since the points X_i and X_i' are defined up to a scale factor due to their projective nature, R_1^2 and $T_{2\to 1}^2$ are defined up to a scale factor. The scale ambiuity implies that one can only determine the unit vector of translation. In addition, given a homography matrix between two images, one can decompose the matrix H to extract the inter-frame rotation and unit vector of translation from the H matrix by using the SVD as follows

$$H = U \Sigma V^T$$

$$\Sigma = diag(\lambda_1, \lambda_2, \lambda_3)$$

where λ_n represents each of the singular values that contain information about the camera motion. When the three singular values are distinct to each other, the decomposition yields four possible mathematical solutions from which just two are physically possible because they represent the motion in front of the camera. The computation of rotation and translation from the singular values is shown below.

$$R_1^2 = U \begin{bmatrix} \alpha & 0 & \beta \\ 0 & 1 & 0 \\ -s\beta & 0 & s\alpha \end{bmatrix} V^T \tag{3.14}$$

$$T_{2\to 1}^2 = \frac{1}{\omega}\left(-\beta u_1 \; + \left(\frac{\lambda_3}{\lambda_2} \; - \; s\alpha\right) u_3\right) \tag{3.15}$$

$$N = \omega(\delta v_1 \; + \; v_3) \tag{3.16}$$

where ω represents a scaling factor so that $||N|| = 1$, and

$$\delta = \pm\sqrt{\frac{\lambda_1^2 - \lambda_2^2}{\lambda_2^2 - \lambda_3^2}}$$

$$\alpha = \frac{\lambda_1 + s\lambda_3\delta^2}{\lambda_2\,(1+\delta^2)}$$

$$\beta = \pm\sqrt{1-\alpha^2}$$

$$s = det(U)det(V)$$

Caballero et al. (2009)

4. Simulation

The quadcopter simulation model used in this thesis was based on the Hector Quadrotor simulation package for the Robot Operating System (ROS) developed by Meyer et al. (2012). In order to test the vision algorithm, the simulation includes a virtual environment made in Gazebo for image rendering and model visualization. A basic explanation of the quadcopter dynamic model is given in this section. Additionally, a description of the simulation environment development with Gazebo and Matlab is provided. Moreover, a second simulation model developed in Simulink by Hartman et al. (2014) was used for the generation of Monte Carlo and visual odometry simulations for the rapid assessment of algorithm performance.

4.1 Quadcopter Dynamic Model

The quadrotor model, as its name suggests, consists of four rotors attached to four different arms attached to a rigid body. Thus the control of the vehicle is achieved by generating a differential thrust and moment generated by each of the rotors. Since the quadrotor can be described as a rigid body, the dynamics of the vehicle can be derived from the rigid body equations of motion by the computation of the sum of forces and moments that are acting on the vehicle.

$$\dot{\vec{P}}^N = \vec{V}^N \tag{4.1}$$

$$m\dot{\vec{V}}^N = mg\vec{e}_3 + R_B^N\vec{F} \tag{4.2}$$

$$J_m\dot{\vec{\Omega}}^B = -\vec{\omega}\times J_m\vec{\Omega} + \vec{\tau} \tag{4.3}$$

where R_B^N is the rotation matrix that transforms a vector from the body B to the navigation N coordinate frame, $\vec{\Omega}$ denotes the angular velocity vector of the body with respect to the navigation frame, J_m and m denote the constant inertia matrix and mass of the rigid body respectively, $\vec{e}_3$ represent a vector of the form $[0,0,1]^T$, and $\vec{P}^N$ and $\vec{V}^N$ represent the position and velocity of the body. In addition, the terms $\vec{F}$ and $\vec{\tau}$ respectively denote the forces and torques applied to the body by the aerodynamics of the rotors (Mahony et al., 2012). A representation of the directions of the applied forces and moments is shown in Figure 4.1.

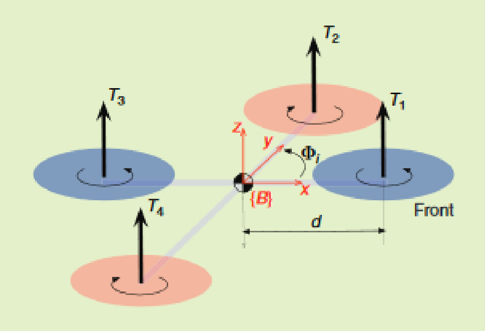


Figure 4.1: Representation of Applied Forces and Torques on the Vehicle (Mahony et al., 2012)

The vertical force (thrust) produced by each of the rotors is modeled using momentum theory as

$$T_i = C_T \; \rho_{air} \, A_{r_i} \; {r_i}^2 \bar{\omega}_i^2 \tag{4.4}$$

where the subindex i represents each of the motors, A_{r_i} represents the disk area of each rotor, r_i denotes the rotor radius, $\bar{\omega}_i$ denotes the angular velocity of each rotor, and ρ_{air} denote the air coefficient given by the standard atmosphere model. Additionally, the coefficient C_T represents the constant thrust coefficient which is dependent on the geometry and profile of the rotor. However, the model can be simplified by grouping the constant values into a single thrust coefficient c_T which should be greater than zero and can be computed experimentally. For the number of propellers $n = 4$, the thrust model is given as:

$$T_i = C_T \; \bar{\omega}_i^2 \tag{4.5}$$

The corresponding summation of forces is given by:

$$F_{\sum} = \sum_{i=1}^{n} |T_i| = C_T \left(\sum_{i=1}^{n} \bar{\omega}_i^2 \right) \tag{4.6}$$

Similarly, the rotation of the blades generates an effect on the yaw due to the force of the propeller. The torque generated by the motor is given by:

$$\tau_Q = C_Q \; \bar{\omega}_i^2 \tag{4.7}$$

where τ_Q denote the torque generated by the motor and C_Q denote the torque coefficient relationship between the motor and the blade (Mahony et al., 2012).

In order to generate a net moment equal to zero, it is necessary to generate counteracting torques by orientating the direction of rotation of the motors in a corresponding clockwise

$(\sigma = +1)$ and a counterclockwise $(\sigma = -1)$ manner. That means that two rotors will have a clockwise rotation and the two remaining rotors a counterclockwise rotation, as shown in Figure 4.1. The moments acting on the quadrotor depend on the arm distance d_i of each rotor to the center of mass of the vehicle. The total sum of moments $\vec{\tau} = \{\tau_\phi, \tau_\theta, \tau_\psi\}$ is described as :

$$\tau_\phi = C_T \sum_{i=1}^{n} \left(d_i\, \bar{\omega}_i^2\right) \tag{4.8}$$

$$\tau_\theta = -C_T \sum_{i=1}^{n} \left(d_i\, \bar{\omega}_i^2\right) \tag{4.9}$$

$$\tau_\psi = C_Q \sum_{i=1}^{n} \left(\sigma_i \bar{\omega}_i^2\right) \tag{4.10}$$

The sum of forces and moments applied to the vehicle can be summarized in a matrix form as

$$\begin{pmatrix} F_\Sigma \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \begin{pmatrix} c_T & c_T & c_T & c_T \\ 0 & dc_T & 0 & dc_T \\ -dc_T & 0 & dc_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{pmatrix} \begin{pmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{pmatrix} \tag{4.11}$$

The matrix described in Eq. 4.11 represents the basic forces and moments that are applied to the body. However, there exist other aerodynamic effects that can affect the model. These effects include blade flapping and aerodynamic induced drag (Mahony et al., 2012). The resultant motion of the vehicle is given by solving Eqs. 4.1 to 4.3 using any numerical integration method for differential equations, such as Runge-Kutta.

4.2 Sensor Modeling

In order to generate an estimation of the quadcopter states, since the states cannot be directly measured, it is necessary to generate a simulated signal of different sensors that can make the quadcopter states observable. Most of the sensing signals, such as the inertial measurements, are derivatives from the dynamic model itself. However, some other signals are dependent on inputs from the dynamic model and also from the simulation environment. The basic models of each sensor are described in this section.

4.2.1 Inertial Measurements

Each of the inertial measurements (accelerometers, gyroscope, magnetometer, barometer) are modeled using an error model described by a first-order Markov process. It is assumed that the state at the next time period is only reliant on the current state of the system and that the noises on each sensor are uncorrelated from the other sensors. The error model of a signal y is thus given as a function of time as:

$$y(t) = \hat{y}(t) + b + \eta_y \tag{4.12}$$

where $\hat{y}(t)$ represents the true value, b is the current bias and η_y is an additive zero-mean white Gaussian noise. In addition, the bias term can either be constant or be subject to a random walk model given as:

$$\dot{b}(t) = \frac{1}{T} b + \eta_b \tag{4.13}$$

where the term T represents a time constant that describes the random drift and η_b denotes an additive zero-mean white Gaussian noise (Meyer et al., 2012).

Inertial Measurement Unit (IMU)

The inertial measurement unit is the sensor that measures the accelerations and angular rates of the vehicle. These signals are referenced with respect to the body reference frame. The IMU model is based on the error model described in Eq. 4.12.

Magnetometer Sensor

The magnetometer sensor that measures the corresponding magnetic field that is acting on the vehicle. The collection of magnetic field information serves as a heading feedback. It is important to remark that the magnetic field magnitude depends on the longitude and latitude coordinates of the body. Therefore, the magnetic field measurements are based on the world magnetic model 2015 (Maus et al., 2015).

Barometer Sensor

A barometer measures the atmospheric pressure to determine the altitude as an absolute measurement independent of terrain elevations or irregularities. This sensor is based on the International Standard Atmosphere (ISA) which describes the pressure, temperature and density of the earth on a sunny nominal day. This model is referenced from the sea level altitude (Meyer et al., 2012).

4.2.2 Camera Model

The basic model of a camera can be approximated by a pinhole projective model, also known as the pinhole camera model, which represents the camera geometric model with a parametrization of the intrinsic and extrinsic camera parameters explained in Section 3.1. The pinhole model projects the points in space onto the camera or image plane. The center of projection corresponds to the origin of an Euclidean coordinate system called *camera center* (Hartley & Zisserman, 2003). The image plane in this model is represented by the focal length as represented in Figure 4.2. Thus by consideration of similar triangles, one can quickly assess the model projection as follows.

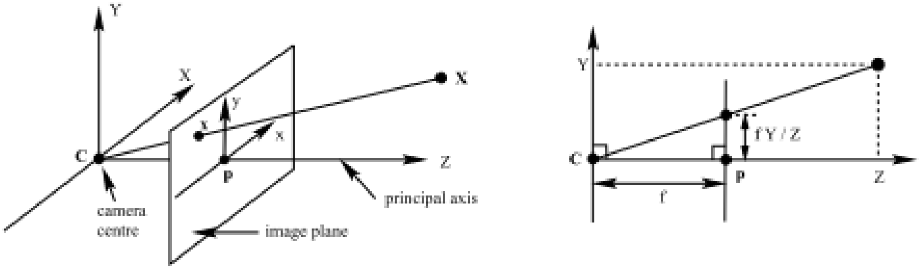


Figure 4.2: Pinhole Camera Model (Hartley & Zisserman, 2003)

$$P = K_{int}R[I|\bar{C}] = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & -R\bar{C} \end{bmatrix} \tag{4.14}$$

where the intrinsic matrix K_{int} is composed of the focal lengths $(f_x,\ f_y)$ and the principal points or camera center $(c_x,\ c_y)$. Additionally, the term R represent the rotation of the camera to the world frame, and $\bar{C}$ represents the coordinates of the camera center in the world coordinates (Hartley & Zisserman, 2003). The projection of a 3D world point $\hat{X}$ in homogeneous coordinates onto the image plane is given by

$$\vec{x}_{cam} = K_{int} R[I|\bar{C}]\hat{X} + \eta_{x,y} \tag{4.15}$$

where the term $\eta_{x,y}$ denotes a zero mean Gaussian noise in the x and y direction in the image plane. The intrinsic parameter matrix used for the simulation is given as:

$$K_{int} = \begin{bmatrix} 374.6706 & 0 & 320.5 \\ 0 & 374.67 & 180.5 \\ 0 & 0 & 1 \end{bmatrix}$$

4.3 Simulation Architecture

Given the ROS simulation model from Meyer & Kohbrecher (2014), the idea is to generate an environment that can be visualized in Gazebo and monitored from Matlab for testing the estimation algorithms and commanding the position of the vehicle. The architecture of this system is summarized in Figure 4.3. Additionally, a detailed tree graph of the ROS topics being published and managed by the ROS master node is shown in Figure 4.4.

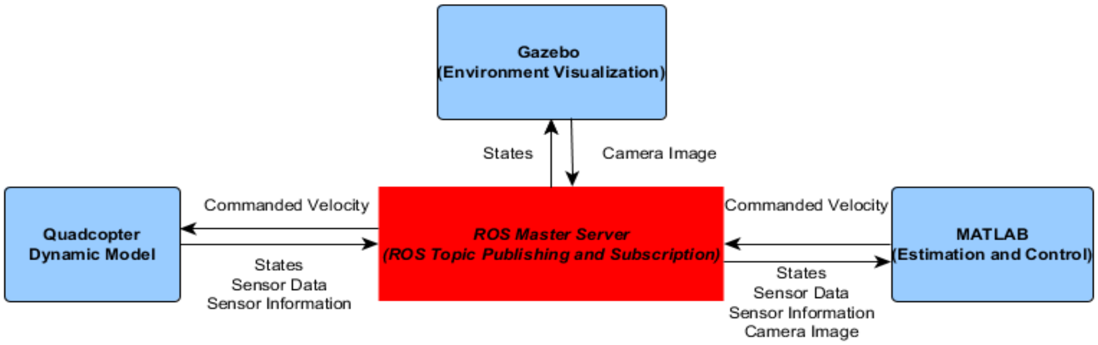


Figure 4.3: Simulation Architecture Diagram

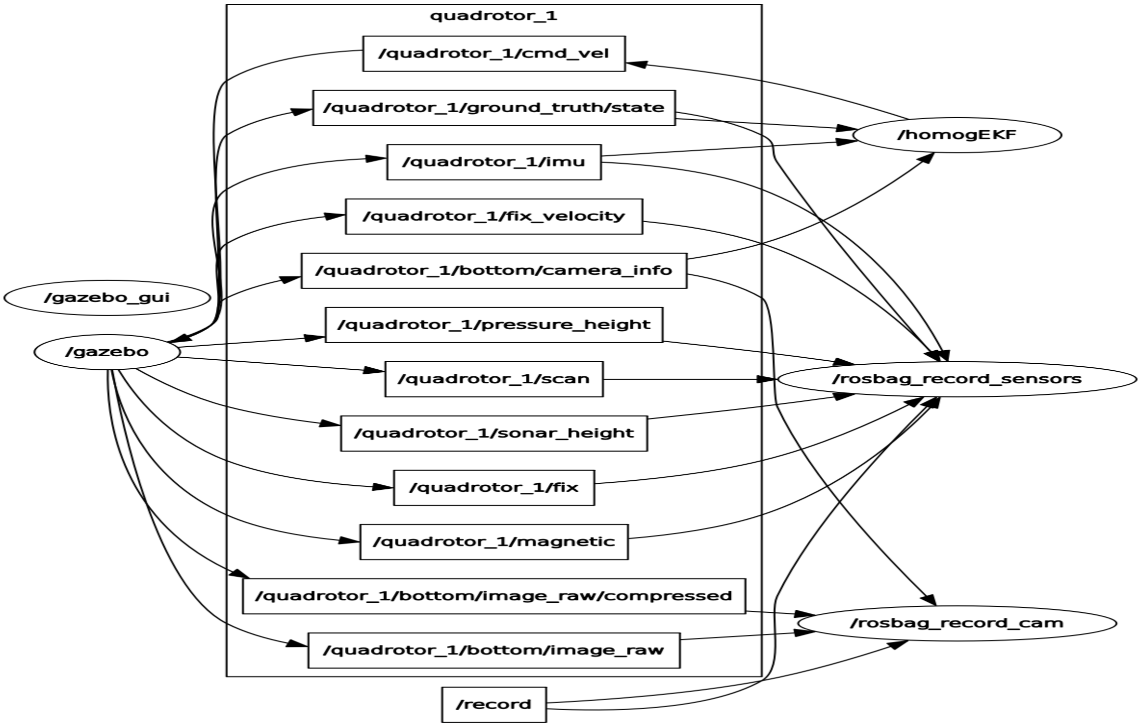

quadrotor_1
/quadrotor_1/cmd_vel
/quadrotor_1/ground_truth/state
/quadrotor_1/imu
/quadrotor_1/fix_velocity
/quadrotor_1/bottom/camera_info
/quadrotor_1/pressure_height
/quadrotor_1/scan
/quadrotor_1/sonar_height
/quadrotor_1/fix
/quadrotor_1/magnetic
/quadrotor_1/bottom/image_raw/compressed
/quadrotor_1/bottom/image_raw
/gazebo_gui
/gazebo
/homogEKF
/rosbag_record_sensors
/rosbag_record_cam
/record

Figure 4.4: Simulation ROS Topic Tree Graph

As it can be seen from Figure 4.3, the system is managed by a central ROS master server that is in charge of the ROS topics shown in Figure 4.4. The Matlab node is in charge of the generation of the commanded velocity that is fed into the quadcopter node to control the vehicle. The Matlab node also contains the estimation of the filters. Additionally, an environment within Gazebo was created using standard objects from the model library in Gazebo. A snapshot of the environment is shown in Figure 4.5.

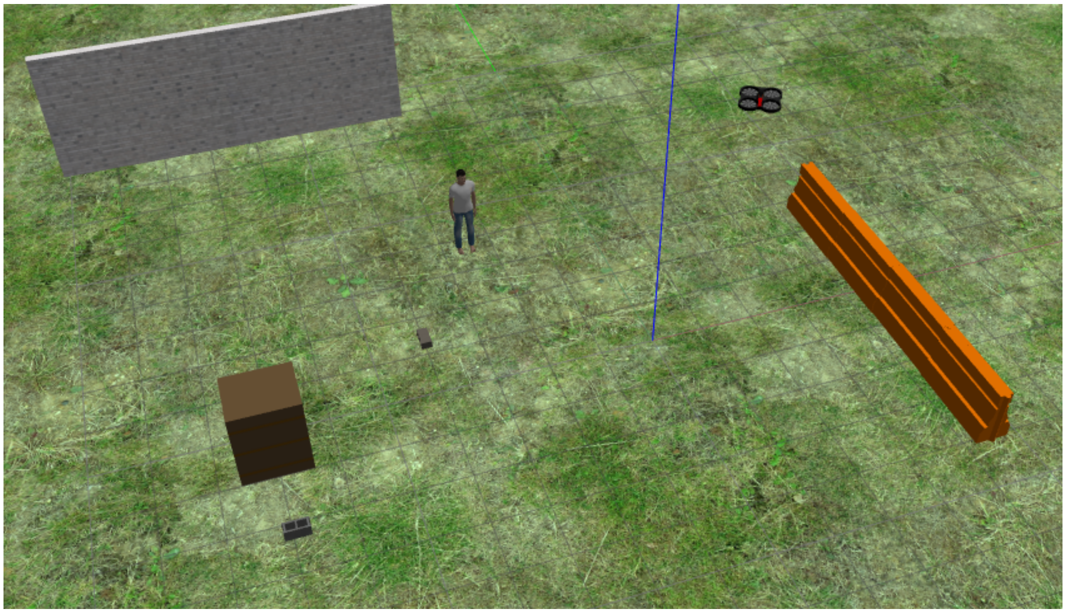


Figure 4.5: Gazebo Environment Visualization

5. Vision-Based State Estimation

In this chapter, a solution for the estimation of the states cor esponding to a 6-DoF vehicle is presented based on the homography measurement. In this approach, a vision only measurement is presented given the computation of the homography matrix H explained in Chapter 3. This approach is known as the dead reckoning solution or the homography visual odometry method. Due to the loss of depth perception, an altitude sensor is used as a source for scaling the measurement. Additionally, the homography matrix measurement is integrated with inertial measurements. This is described in Section 5.2 as a temporary solution for the characteristic drift present in a dead reckoning solution. Finally, some simulation results are presented together with an analysis of the performance of each algorithm using a Monte Carlo Simulation.

The following coordinate systems are used for the development of the estimation filter:

1. Navigation (Earth-Referenced) Frame N: Local north-east-down frame in which the X_N and Y_N axes are aligned with the north and east magnetic poles of the earth respectively, and the Z_N axis is pointed downward.

2. Body-Fixed Frame B: Body-fixed reference frame in which the origin is located at the center of mass of the vehicle, the body-x axis points towards the front of the vehicle, the body-y axis points out the right side of the vehicle, and the body-z axis is pointed downward relative to the vehicle.

3. Camera-Fixed Frame C: Camera-fixed reference frame that is aligned with the axes of the camera. For simplicity and without loss of generality, in this document, the camera-fixed reference frame is assumed to be located at the center of mass of the vehicle at a fixed angle ϑ with respect to the horizontal plane of the vehicle, and a fixed angle γ with respect to the vertical plane of the vehicle.

5.1 Homography-Based Visual Odometry

Consider a quadcopter carrying a monocular camera located at the center of mass of the vehicle looking straight downward, as shown in Figure 5.1. The pose of the vehicle can be estimated by computing the homography matrix between camera measurements at two instances of time and then extracting the information about the motion between the frames. The homography relationship is ideal for the landing scenario since quadcopters usually require a planar landing surface. Therefore, the homography matrix can be used to extract the pose information by means of vision only measurements. In this case, the feature points seen by the camera are the corners features of the landing pad.

As a first step, it is important to be able to robustly track feature points in the images. Algorithms such as the Kanade-Lucas-Tomasi (KLT) or Harris corner detectors, features from accelerated segment test (FAST), or Speeded-Up Robust Features (SURF) can be used to detect and track feature points. However, it is important to filter outliers before the computation of the homography matrix. After removing outliers, the previously described DLT can be used to compute an initial estimation of the homography matrix. After computing

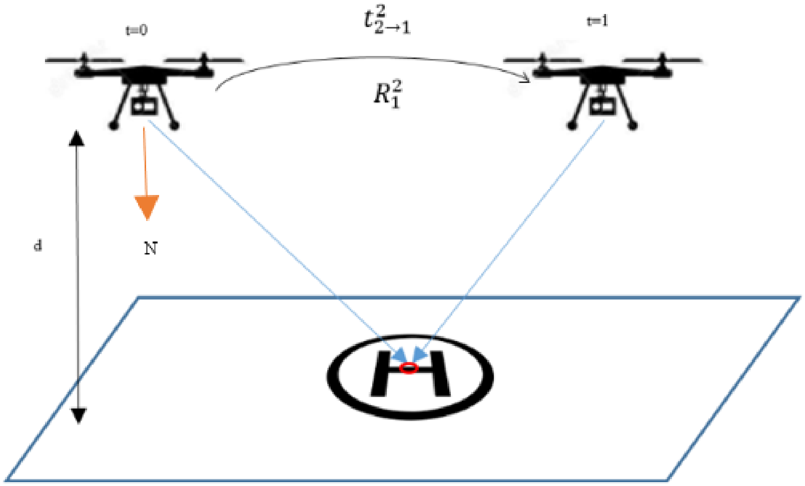


Figure 5.1: Two-Frame Homography Over a Planar Set of Points

the estimation, it is necessary to generate a robust homography estimation using the steps described in Algorithm 3.1.

Once the homography matrix has been computed, the rotation and translation information of the camera can be extracted using the decomposition methods described in Section 3.2.4. However, when the homography matrix is decomposed, the algorithm returns four possible solutions. Two solutions can be eliminated by imposing the positive depth constraint shown below

$$N_z > 0$$

The positive depth constraint means that solutions with negative plane normals are discarded based on the third component of the normal solutions that the homography provides. However, after applying the positive depth constraint, there are still two different possible solutions, known as the possible physical solutions, that need to be analyzed. In order to disambiguate this solution, one can use previous information about the normal or compare each of the extracted normals with a second set of normals using the homography about the same frame at another instant of time.

5.1.1 Previous Normal Comparison

Given the decomposition of the homography matrix H, from frame 1 to frame 2, a set of two solutions containing the two physical solutions for translation and rotation are represented as

$$S_1 = \begin{Bmatrix} R^2_{11} \\ T^2_{2\to 11} \\ N_1 \end{Bmatrix} \qquad S_2 = \begin{Bmatrix} R^2_{12} \\ T^2_{2\to 12} \\ N_2 \end{Bmatrix} \tag{5.1}$$

It is important to recall that the normal that is extracted from the homography decomposition is the unit vector normal of the feature plane resolved in the camera reference frame, which in this case is aligned with the body reference frame. Therefore, if the last GPS measurement is available or if the quadcopter is at a horizontal position relative to the ground such that the N vector is $N_{quad} = (0,0,1)^T$, one can use this information to disambiguate the set of solutions from Eq. 5.1 by comparing the normals N_1 and N_2 with the normal N_{quad} since the real solution normal should coincide with the normal of the quadcopter. However, the normal from the decomposed homography will be corrupted by noise from the camera measurements. Therefore, one of the solutions will not always exactly coincide with the normal of the quadcopter. For this reason, the following metric is used:

$$\hat{S}_i = \begin{cases} \delta_1 = & \|N_1 - N_{quad}\| \in S_1 \\ \delta_2 = & \|N_2 - N_{quad}\| \in S_2 \end{cases} \tag{5.2}$$

where δ_i represents the norm of the difference between the normal N_i and the real normal provided by the initial quadcopter measurement. The real solution is extracted from the solution set $\hat{S}_i$ as follows

$$S = min\{\delta_1, \delta_2\} \tag{5.3}$$

Once the difference has been computed, the selected solution corresponds to the lowest difference δ_n as explained in Eq. 5.3. This will provide the first inter-frame rotation and translation estimate. In order to propagate the solution, it is possible to rotate the normal given the camera inter-frame rotation as

$$N_{quad}(k) = R_1^2 N_{quad}(k-1) \tag{5.4}$$

where $N_{quad}(k)$ represents the propagated normal of the quadcopter, $N_{quad}(k-1)$ represents the previous normal solution, and R_1^2 denotes the disambiguated rotation solution.

5.1.2 Multiple Homography Comparison

For cases in which a first measurement or estimation of the normal of the quadcopter at time $t = 0$ is not available, it is possible to generate a disambiguation by comparing several decomposed normals coming from the decomposition of the homography matrix of a third frame with respect to the first one. A representation of this case is provided in

Figure 5.2, where a second homography with respect to the first frame is computed just for disambiguation purposes. This approach represents a modification of the one taken in (Caballero et al., 2009).

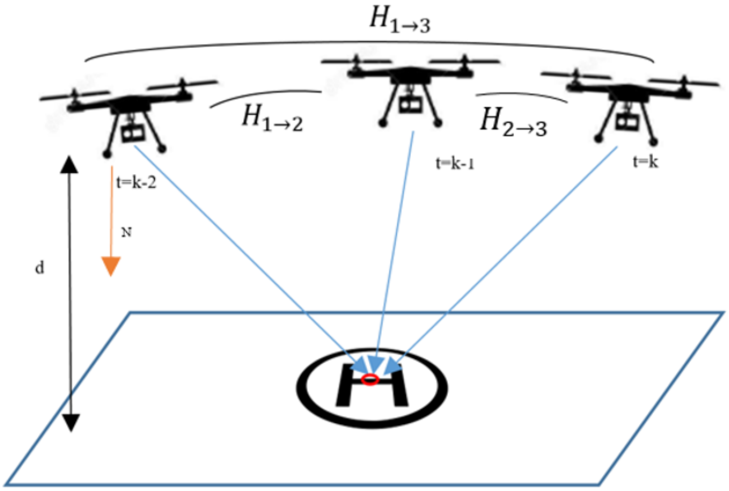


Figure 5.2: Three-Frame Homography Over a Planar Set of Points

Decomposing the homography matrices $H_{1\to2}$ and $H_{1\to3}$ yields a set of four solutions from which just two are used for the posterior pose reconstruction. This means that from the set of solutions coming from the matrix $H_{1\to3}$, just the computed normals are used. The set described in Eq. 5.1 still holds for this method. Then we have that the metrics defined for this comparison are given with respect to the set of normals described in Eq.5.5 as follows

$$N_{13_i} = \{N_{13_1}, N_{13_2}\} \tag{5.5}$$

In order to disambiguate the solution, the following comparison is made to generate the solution set $\hat{S}_i$

$$\hat{S}_i = \begin{cases} \delta_{1_1} = & \|N_1 - N_{13_1}\| \in S_1 \\ \delta_{1_2} = & \|N_1 - N_{13_2}\| \in S_1 \\ \delta_{2_1} = & \|N_2 - N_{13_1}\| \in S_2 \\ \delta_{2_2} = & \|N_2 - N_{13_2}\| \in S_2 \end{cases} \tag{5.6}$$

Similar to Eq: 5.3, the solution set $\hat{S}_i$ is used in Eq. 5.7 to extract the disambiguated solution.

$$S = min\{\delta_{1_1}, \delta_{1_2}, \delta_{2_1}, \delta_{2_2}\} \tag{5.7}$$

where δ_{i_j} denotes the norm of the difference between the normals from the homography $H_{1\to 2}$ and the set of normals N_{13_i} computed from $H_{1\to 3}$. Then the solution is given by the minimum value of δ_{i_j}.

It should be noted that this comparison can be made using at least two homographies with respect to the frame under analysis. However, for real-time implementations, this method represents an increment in the delay of the measurements which should be taken into account when the estimate is used as a feedback signal for camera servo control applications.

5.1.3 Pose Reconstruction

Once the homography has been decomposed and disambiguated, it is necessary to relate this interframe motion with the inertial attitude and pose of the UAV. However, when the monocular vision approach is used, scale (range) information is lost in the 2D projection into the camera frame. Thus, the translation vector $T^2_{2\to 1}$ is only defined up to a scale factor. In order to recover the scale, a measurement of the distance d from the camera to the feature point plane is required, as shown in Figures 5.1 and 5.2. This information could be provided using a sensor that provides a direct altitude measurement, such as a barometer. A stereo vision measurement system could also be used to extract range information; however, the range of the measured distance is limited by the baseline distance between the two cameras.

Denote ${R_B}^N(k)$ as the Direction Cosine Matrix (DCM) representing the rotation from the body reference frame (B) to the navigation reference frame (N) when the actual frame occurs, and $R_B^N(k-1)$ the DCM rotation matrix from B to E when the previous frame occurs. Additionally, the matrix R_C^B represent the constant rotation matrix from the camera reference frame C to the body reference frame B. Thus the homography measurements coming from the decomposition of Eq. 3.13 can be related to the pose of the quadcopter as

$$R_{C_{k-1}}^{C_k}(k) = R_C^{B^T} R_B^{N^T}(k) R_B^N(k-1) R_C^B \tag{5.8}$$

$$T_{C_k \to C_{k-1}}^{C_k}(k) = R_C^{B^T} R_B^{N^T}(k)(P^N(k-1) - P^N(k)) \tag{5.9}$$

$$N(k) = R_C^{B^T}\ R_B^{N^T}(k-1)\ e_3 \tag{5.10}$$

$$d(k) = {e_3}^T(P^N(k-1)) \tag{5.11}$$

where P^N represents the inertial position of the quadcopter in the navigation reference frame, $R_{C_{k-1}}^{C_k}$ represent the rotation from the previous frame $(k-1)$ to the current frame (k). Additionally, e_3 denotes a vector of coordinates $e_3 = (0,0,1)^T$. Finally, $T_{C_k \to C_{k-1}}^{C_k}$ represents the translation from the current frame (k) to the previous frame $(k-1)$ with respect to the current frame.

In order to compute the current pose of the vehicle at time k, one can manipulate Eqs. 5.8 → 5.11 to generate a daisy chained approach for the generation of the following expressions for attitude and relative position respectively (Kaiser et al., 2010).

$$T_{rB}^{N}(k) = \left(T_{rB_1}^{B_2}\right)^{-1} \left(T_{rB_2}^{B_3}\right)^{-1} \dots \left(T_{rB_{k-1}}^{B_k}\right)^{-1} \tag{5.12}$$

where T_r is the homogeneous representation of the rotation matrix $R_{C_{k-1}}^{C_k}$ and scaled translation $T_{C_k \to C_{k-1}}^{C_k}$ expressed as

$$T_{rB_{k-1}}^{B_k} = \begin{bmatrix} R_C^B \, R_{C_{k-1}}^{C_k}(k) & d(k) \, T_{C_k \to C_{k-1}}^{C_k}(k) \\ \bar{0} & 1 \end{bmatrix} \tag{5.13}$$

where d is used as the scaling factor measurement coming from the barometer sensor at each time step. Then the attitude $\rho(k)$ (Euler angles) can be extracted from the DCM matrix as:

$$\rho(k) = \begin{cases} \phi(k) = \arctan\left(\dfrac{R_{B_k}^{N}(3,2)}{R_{B_k}^{N}(3,3)}\right) \\ \theta(k) = -\arcsin\left(R_{B_k}^{N}(1,3)\right) \\ \psi(k) = \arctan\left(\dfrac{R_{B_k}^{N}(1,2)}{R_{B_k}^{N}(1,1)}\right) \end{cases} \tag{5.14}$$

5.2 Homography-Based Kalman Filter

While the homography relationship can be used in the form of visual odometry, a vision-only solution for the computation of the motion of a vehicle can rapidly accumulate drift errors that affect the estimation as explained in Section 5.1. It is true that a covariance can be included as part of drift correction in the visual odometry solution. However, without the addition of more information about the environment, the uncertainty of the prediction model remains high. Therefore, a more robust approach for uncertainty bounding is the inclusion of inertial measurements into the navigation prediction model. Such an approach can also address the scale ambiguity associated with monocular vision. However, the inclusion of more information yields the addition of different sources of noise that need to be taken into account.

In this thesis, the proposed sensor fusion is done through the implementation of an extended Kalman filter (EKF) model that integrates accelerometer and rate gyro data from the onboard IMU as part of the state propagation phase of the filter. The measurement update in the filter is based on the reconstruction of the nine parameters of the homography matrix since the disambiguation on the decomposition can lead to false fixes as shown in Section 5.1. However, due to the loss of depth perception, an altitude measurement is needed to reconstruct the homography. A diagram of the proposed architecture of the filter is shown in Figure 5.3 where the depth reconstruction is done using a barometer or a laser range finder. Additionally, a magnetometer measurement is added to the system as a means of correcting the heading drift.

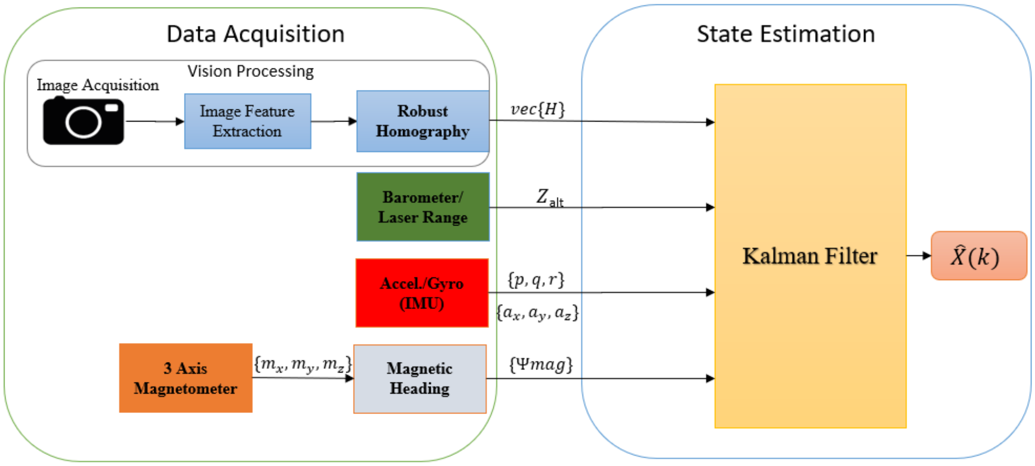


Figure 5.3: EKF Architecture Diagram

5.2.1 Process Model

Since the Kalman filter is designed to work with additive zero-mean Gaussian noise, it is assumed that all the sensors, including the camera measurements, are corrupted by zero-mean Gaussian noise that obeys a first order Markov process. As a mean of correction of the errors in the IMU (accelerometers, and gyroscope), the IMU bias is added to the states to be estimated. The state estimation is composed of 9 states (position, velocity, and orientation) from the 6-DoF UAV plus the addition of 6 states from the IMU bias estimation, making the EKF a 15^{th} order filter. The estimation vector is described by Eq. 5.15 as:

$$\hat{X} = \left\{P^N,\ V^N,\ \rho,\ b_{acc},\ b_{gyr}\right\}^T \tag{5.15}$$

where P^N and V^N represent the position and velocity in the navigation reference frame respectively, $\rho = \{\phi, \theta, \psi\}$ represent the attitude Euler angles. Similarly, b_{acc} denote the three-axis accelerometer bias estimation and b_{gyr} represent the gyroscope bias estimation.

The process is formulated as part of a non-linear discrete system where the time size Δt_s is defined by the IMU since it runs at the highest frequency and it is used as part of the process propagation. The corresponding process propagation model can be described as a function of the process measurement input u and the associated process measurement random Gaussian white noise w associated with the model. This is modeled in Eq. 5.16 as:

$$X_k = \Phi(X_{k-1},\ u_k + w_k) \tag{5.16}$$

The full process model is defined as follows:

$$\begin{Bmatrix} P^N(k) \\ V^N(k) \\ \rho(k) \\ b_{acc}(k) \\ b_{gyr}(k) \end{Bmatrix} = \begin{bmatrix} P_N(k-1) + V_N(k-1)\Delta t_s \\ V_N(k-1) + (R_B^N(k-1)a^b(k))\Delta T_s \\ \rho(k-1) + (L_B^N(k-1)\omega^B(k))\Delta T_s \\ b_{acc}(k-1) \\ b_{gyr}(k-1) \end{bmatrix} \tag{5.17}$$

where a^B denotes the acceleration in the body reference frame measured by the accelerometer. The term ω^B represents the angular acceleration in body reference frame coordinates measured by the gyroscope. Additionally, R_B^N represents the DCM rotation matrix that transforms the coordinates from the body reference frame to the navigation reference frame. The term L_B^N denotes the Euler angle kinematics relationship given by

$$L_B^N = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \tag{5.18}$$

The IMU measurements used in the process model are known to be corrupted by slow-varying sensor biases and white noise w_n. Thus the model of IMU measurements is given as

$$\begin{aligned} a^B_{acc} &= a^B - b_{acc} - w_{acc} \\ \omega^B_{gyr} &= \omega^B - b_{gyr} - w_{gyr} \end{aligned} \tag{5.19}$$

When linearized around the previous estimate, the process model takes the form of a discrete state space system given by

$$X_k = F_{k-1}\; X_{k-1} + G_{k-1}\; u_(k-1)$$

where u_k represents the IMU measurement input that is associated to the white noise w_k as modeled in Eq. 5.19. F_k, G_k, and u_k matrices are given by

$$\begin{gathered} F_{k-1} = \begin{bmatrix} I_{3\times3} & I_{3\times3}\Delta t_s & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & \dfrac{\partial R^N_B}{\partial \rho} & -R^N_B \Delta t_s & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} + \dfrac{\partial L^N_B}{\partial \rho} & 0_{3\times3} & -L^N_B \Delta t_s \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix} \\ G_{k-1} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ R^N_B \Delta t_s & 0_{3\times3} \\ 0_{3\times3} & L^N_B \Delta t_s \\ 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \qquad u_{k-1} = \begin{bmatrix} a^B_{acc} \\ \omega^B_{gyr} \end{bmatrix} \end{gathered} \tag{5.20}$$

5.2.2 Measurement Model

Based on the derivation described in Zhao et al. (2012), the measurement vector $\hat{Z}$ takes the form of:

$$\hat{Z}_k = \left\{ H_{vec}(k),\ -Z_{baro}(k),\ \psi_{mag}(k) \right\}^T \tag{5.21}$$

where the barometer is used as the altitude measurement and is represented as Z_{baro}, and the term ψ_{mag} represents the magnetic heading. Additionally, $H_{vec}(k)$ corresponds to the normalized 9 entries of the homography matrix, stacked into a vector form.

The homography measurement is related to the vehicle states by approximating the Eqs. 5.8 → 5.11 as follows:

$$R^{C_k}_{C_{k-1}}(k) = {R^B_C}^T {R^N_B}^T(k) R^N_B\left(k - \frac{\Delta t_s}{\Delta t_v}\right) R^B_C \tag{5.22}$$

$$T^{C_k}_{C_k \to C_{k-1}}(k) \approx -{R^B_C}^T {R^N_B}^T(k) V_N(k) \Delta t_v \tag{5.23}$$

$$N(k) \approx {R^B_C}^T\ {R^{C_k}_{C_{k-1}}(k)}^T {R^N_B}^T(k)\ e_3 \tag{5.24}$$

$$d(k) \approx -{e_3}^T (P^N(k) - V^N(k)\Delta t_v) \tag{5.25}$$

where Δt_v represents the time step size at which the vision system provides a measurement. Additionally, the rotation matrix $R^N_B\left(k - \frac{\Delta t_s}{\Delta t_v}\right)$ is a a function of $\rho(k)$, but it is related to the previous attitude estimate at time $t = \left(k - \frac{\Delta t_s}{\Delta t_v}\right)$. Based on the measurements modeled in Eqs. 5.22 → 5.25, the homography matrix is reconstructed using Eq. 3.13, stacked in a vector form, and then normalized. The reconstructed measurement then takes the form described in Eq. 5.26 and Eq. 5.27.

$$H_{proj}(k) = R^{C_k}_{C_{k-1}}(k) + \frac{T^{C_k}_{C_k \to C_{k-1}}(k)}{d(k)} N(k)^T \tag{5.26}$$

$$H_{vec}(k) = vec\left(\frac{H_{proj}(k)}{H_{proj_{3,3}}(k)}\right) \tag{5.27}$$

where the operator *vec* represent the column vector stacking. Moreover, the equation that defines the magnetic heading from the three-axis magnetometer is:

$$\psi_{mag}(k) = \arctan_2 \left(\frac{-m^B_y \cos(\hat{\phi}(k)) \; + \; m^B_z \sin(\hat{\phi}(k))}{m^B_x \cos(\hat{\theta}(k)) \; + \; m^B_y \sin(\hat{\phi}(k)) \sin(\hat{\theta}(k)) \; + \; m^B_z \cos(\hat{\phi}(k)) \sin(\hat{\theta}(k))} \right) \tag{5.28}$$

given that $\{m^B_x,\, m^B_y,\, m^B_z\}$ are each the magnetic readings from the magnetometer measured with respect to the body reference frame of the vehicle and expressed in microTeslas μT. The terms $\hat{\phi}$ and $\hat{\phi}$ are the last estimates of roll and pitch respectively.

The measurement model is given by the following equation:

$$Z_k = h(X_k) + n_k$$

where n_k represents the corresponding noise associated with the measurement. Moreover, the linearization of the measurement model is given by the Jacobian of $h(X)$ with respect to X, given by

$$H_k = \begin{bmatrix} \frac{\partial H_{vec}}{\partial P_N}_{9\times3} & \frac{\partial H_{vec}}{\partial V_N}_{9\times3} & \frac{\partial H_{vec}}{\partial \rho}_{9\times3} & 0_{9\times3} & 0_{9\times3} \\ e_3^T e_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & e_3^T e_3 & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{5.29}$$

5.2.3 Observability Analysis

In this section a numerical model for the analysis of observability of the proposed navigation system is provided. The observability analysis provides a measure of performance during the design of the algorithm as well as a measure of the observability of the states. When the condition of observability is guaranteed, the asymptotic stability of the error system in the Kalman filter is assured (Lewis et al., 2007).

Consider a discrete stochastic linear system of the form

$$\begin{aligned} x_k &= \Phi_{k,0}\, x_0 \\ z_k &= H_k\, x_k + v_k \end{aligned} \tag{5.30}$$

where $x_k \in \mathbb{R}^n$ represents the state vector at time step $t_k = k\Delta_t$, $x_0 \in \mathbb{R}^n$ represents the initial state vector as a normal random variable with mean $\bar{x}_0$, and $\Phi_{k,0} \in \mathbb{R}^{n\times n}$ represents the state transition matrix spanned from time step $t_k = 0$ to time step $t_k = k\Delta_t$. Similarly, $z_k \in \mathbb{R}^m$ represents the measurement vector at time step $t_k = k$, $H_k \in \mathbb{R}^{m\times n}$ represents the measurement matrix at time step $t_k = k\Delta_t$, and $v_k \in \mathbb{R}^m$ represents a measurement noise vector with Gaussian distribution, zero mean, and covariance R_k.

Let P_0 represent the process covariance matrix at time step $t_k = 0$. Likewise, let $\hat{y}_k \in \mathbb{R}^m$ represent the innovation vector given as $\hat{y}_k = z_k - H_k\hat{x}_k$. Then, the optimal estimation problem obeys the Riccatti equation solution that minimizes the following cost function:

$$J = (\bar{x}_0 - x_0)^T P_0^{-1} (\bar{x}_0 - x_0) + \sum_{k=0}^{N} \left(\hat{y}_k^T R_k \hat{y}_k\right) \tag{5.31}$$

Thus the optimal state estimate is given as

$$\hat{x}_{0,k} = \left(P_0^{-1} + \mathcal{L}_{0,k}\right)^{-1} \left(K_{0,k} + P_0^{-1}\bar{x}_0\right) \tag{5.32}$$

where $K_{0,k}$ represents the observability gain of the form

$$K_{0,k} = \sum_{k=0}^{N} \left(\Phi_{k,0}^T H_k^T R_k^{-1} y_k\right) \tag{5.33}$$

Similarly, $\mathcal{L}_{0,k}$ is the observability gramian, also known as the Fisher information matrix, for a nondeterministic system and is given as:

$$\mathcal{L}_{0,k} = \sum_{k=0}^{N} \left(\Phi_{k,0}^T H_k^T R_k^{-1} H_k \Phi_{k,0}\right) \tag{5.34}$$

(Hong et al., 2008)

It is said that the system is stochastically observable on the time span $T = [0,\ N]$, if for every integer N the observability gramian satisfies the condition

$$\alpha_0 I \leq \mathcal{L}_{0,k} \leq \alpha_0 I \tag{5.35}$$

for some $N > k$, and $0 > \alpha_0 > \alpha_1$. This observability condition guarantees that for a large k, the behavior of $P_{0,k}$ is unique independent of P_0 (Lewis et al., 2007). However, if the system is unobservable, then there exist an unobservable state vector x_u, such that $\mathcal{L}_{0,k} x_u = 0$. Thus the unobservable subspace corresponds to the null space of $\mathcal{L}_{0,k}$. Therefore, the observability analysis requires a rank test on the observability gramian matrix.

Let the SVD of the observability gramian be given as:

$$\mathcal{L}_{0,k} = U_k \Sigma_k U_k^T \tag{5.36}$$

where $U_k = [U_{ko} \quad U_{ku}]$ and $\Sigma_k = \begin{bmatrix} \Sigma_{ko} & 0 \\ 0 & 0 \end{bmatrix}$, given that the singular values Σ_{ko} span the observable subspace composed by observable singular vectors U_{ko} and that the zero singular values span the nullspace provided by the unbservable singular vectors U_{ku}. Thus the singular values provide a rank metric on the observability of the system (Hong et al., 2008).

In order to study the degree of observability of the observable subspace, it is necessary to examine the behavior of the process error covariance matrix $P_{0,k}$ (Hong et al., 2008). Let the process error covariance matrix be befined as

$$P_{0,k} \triangleq E\left[\left(\hat{x}_{0,k} - x_0\right)\left(\hat{x}_{0,k} - x_0\right)^T\right] \tag{5.37}$$

It can be shown that the relationship between the covariance and the observability gramian is given as

$$(P_0 - P_{0,k})P_0^{-1} = P_{0,k}\mathcal{L}_{0,k} \tag{5.38}$$

It can be inferred from the relationship in Eq. 5.38 that the null space of $(P_0 - P_{0,k})$ corresponds to the unobservable subspace. However, since the null spaces for $\mathcal{L}_{0,k}$ and $(P_0 - P_{0,k})$ can differ, the nullspace spanned by $(P_0 - P_{0,k})u = 0$ is going to be called the unestimable subspace. The unestimable subspace tends to go in the direction where the error covariance is relative small, thus generating a possible discrepancy between the tendencies in the directions of convergence of the observable and unestimable subspaces.

A system is called estimable on the time span $T = [0, N]$ if $(P_0 - P_{0,k}) > 0$. Thus, the nullspace of $(P_0 - P_{0,k})$ corresponds to the unestimable subspace. Therefore, the degree of estimability of the observable system can be given by the relative change of the standard deviation of the errors which is influenced by the selection of the initial error covariance matrix (M. H. Lee et al., 2012). Thus, from Eq. 5.38, the rate of change of the error covariance matrix *nu* can be defined as:

$$\nu = \left(\sqrt{P_0} - \sqrt{P_{0,k}}\right)\left(\sqrt{P_{0,k}}\right)^{-1} = 1 - \frac{1}{\sqrt{P_0(i,i)\mathcal{L}_{0,k}(i,i)+1}} \tag{5.39}$$

Thus, for some $1 \leq i \leq n$, if $P_0(i,i)\mathcal{L}_{0,k}(i,i) \ll 1$, then the rate of change of the the standard deviation of the *i*th component is almost zero. If the rate of change is almost zero, it can be assumed that the *i*th component is almost unestimable (M. H. Lee et al., 2012). Therefore, by analyzing the relative standard deviation change of the *i*th component from the left side of side of Eq. 5.39 one can determine the degree of estimability of the stochastic system (M. H. Lee et al., 2012). Consequently, if the span of a vector x_u is unobservable ($\mathcal{L}_{0,k}x_u = 0$), then the span of $P_0x_u = 0$ is unestimable. Therefore, if $\nu < \nu^*$ for a vector u given a threshold ν^*, then the subspace is considered to be weakly estimable for P_0 and consequently the span P_0u can be assumed to be weakly observable (Hong et al., 2008).

5.2.4 Extended Kalman Filter Model

The nonlinear system dynamic model can be written in discrete-time as

$$\hat{x}_k^- = \hat{x}_{k-1}^+ \; + \; \Phi(\hat{x}_{k-1}^+, \; w_k)\Delta t_s \tag{5.40}$$

which is known as the *a priori* state estimate. Because of the nonlinear nature of Eq.5.30, the EKF model approach requires the model to be a first-order Taylor-Mac Laurin expansion with respect to the previous estimate $\hat{x}^+_{k-1}$. This model is obtained by the computation the Jacobian matrix given as:

$$F_k = \left.\frac{\partial \Phi(x)}{\partial x}\right|_{x=\hat{x}^+_{k-1}}$$

Using the computed Jacobian, the model expressed in Eq. 5.30 can be expressed as a linear model without affecting the estimation, given that the propagation interval Δt_s is sufficiently small for the approximation. Thus the linear model takes the form of:

$$x_{k+1} = F_k\, x_k + G_k\, w_k \tag{5.41}$$

In order to compute the uncertainty of the system estimation, it is necessary to compute an *a priori* error covariance using the linearization model in Eq. 5.31

$$P^-_k = F_{k-1} P^+_{k-1} F^T_{k-1} \; + \; G_k Q_{k-1} G^T_k \tag{5.42}$$

where Q represents the error covariance matrix computed from the variances in the process model. This is sometimes the tuning parameter for the estimation.

The measurement model of the EKF is given as a nonlinear function of the state vector $h(x)$ and is defined as

$$z_k = h(\hat{x}^-_k) + n_k \tag{5.43}$$

In order to generate the estimation it is necessary of obtaining an observability gain referred to as the Kalman gain. The Kalman gain takes the following form:

$$K_k = P^-_k H^T_k \left(H_k P^-_k H^T_k + R_k\right)^{-1} \tag{5.44}$$

where H_k and R_k represent the measurement linearization Jacobian matrix and the error covariance matrix of the measurement model respectively.

As part of the estimation, the EKF uses the measurement model to update the state vector (*a posteriori* state estimation). This is accomplished by defining the measurement innovation as:

$$\begin{aligned} \hat{x}_k^+ &= \hat{x}_k^- + K_k \left[z_k - h(\hat{x}_k^-) \right] \\ &= \hat{x}_k^- + K_k \delta z_k^- \end{aligned} \tag{5.45}$$

As a final step in the EKF, it is necessary to recompute and correct the error covariance matrix with the new information computed from Eq. 5.45. The *a posteriori* covariance matrix is computed using the Kalman gain as:

$$P_k^+ = P_k^- \ - \ K_k H_k P_k^- \tag{5.46}$$

Note that the propagation model and the measurement model are running at different frequencies, given that the frequency of the propagation model is defined by the IMU and that the frequency of the measurement model is defined by the frequency of the vision system. The innovation component of the EKF is run at the frequency of the vision system. This means that in between updates, the estimation is defined as a dead reckoning solution.

5.3 Simulation Results

In this section, the visual odometry and the homography-based filter are tested using the simulation models described in Chapter 4. The estimation models are tested using the simulated sensor parameters shown in Table 5.1.

Table 5.1: Simulation sensor parameters

Sensor	*Accelerometer*			*Gyroscope*			*Magnetometer*			*Camera*	*Barometer*
	a_x	a_y	a_z	p	q	r	m_x	m_y	m_z	$\vec{x}_{cam}$	Z_{baro}
Units	m/s^2			deg$/s$			μT			*pixel*	m
Std. Dev. (σ)	0.356	0.6498	0.3846	0.022	0.0208	0.029	0.000169	0.000169	0.000169	0.007	0.238
Bias (b)	0.044	-0.0022	0.071	-0.0028	0.005	0.00154	0	0	0	-	0
Sample Rate	$25Hz$			$25Hz$			$5Hz$			$5Hz$	$5Hz$

5.3.1 Visual Odometry Simulation Results

A visual odometry solution is tested using the Previous Normal comparison and the Multiple Homography comparisons. For this simulation, a landing target with exactly four corners was projected onto the simulated camera plane. These corners were used as a source for the computation of the homography matrix. Additionally, it was assumed that the vehicle was orbiting the landing target and that the target was not lost from the line of sight of the camera at any point of the simulation. The given simulated trajectory is shown in Figure 5.9.

The visual odometry algorithm solution was tested with a pixel noise variance of 0.007 pixels per frame. A comparison between each of the disambiguation algorithms is presented in Figure 5.5 and 5.6 for position and attitude respectively together with their respective error plot on the right side of each plot.

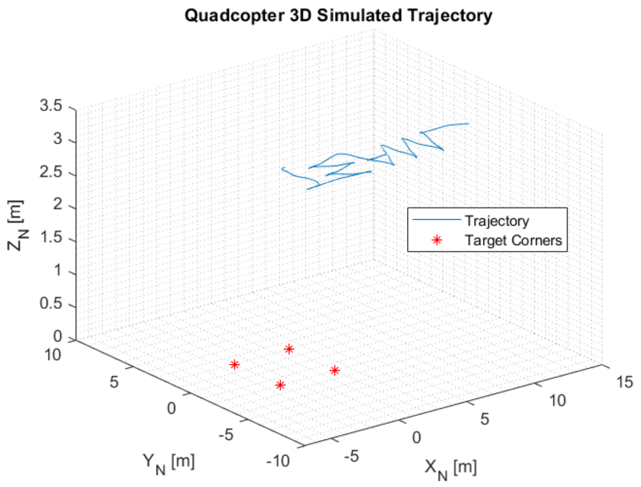


Figure 5.4: Quadcopter Simulated 3D Trajectory for Visual Odometry

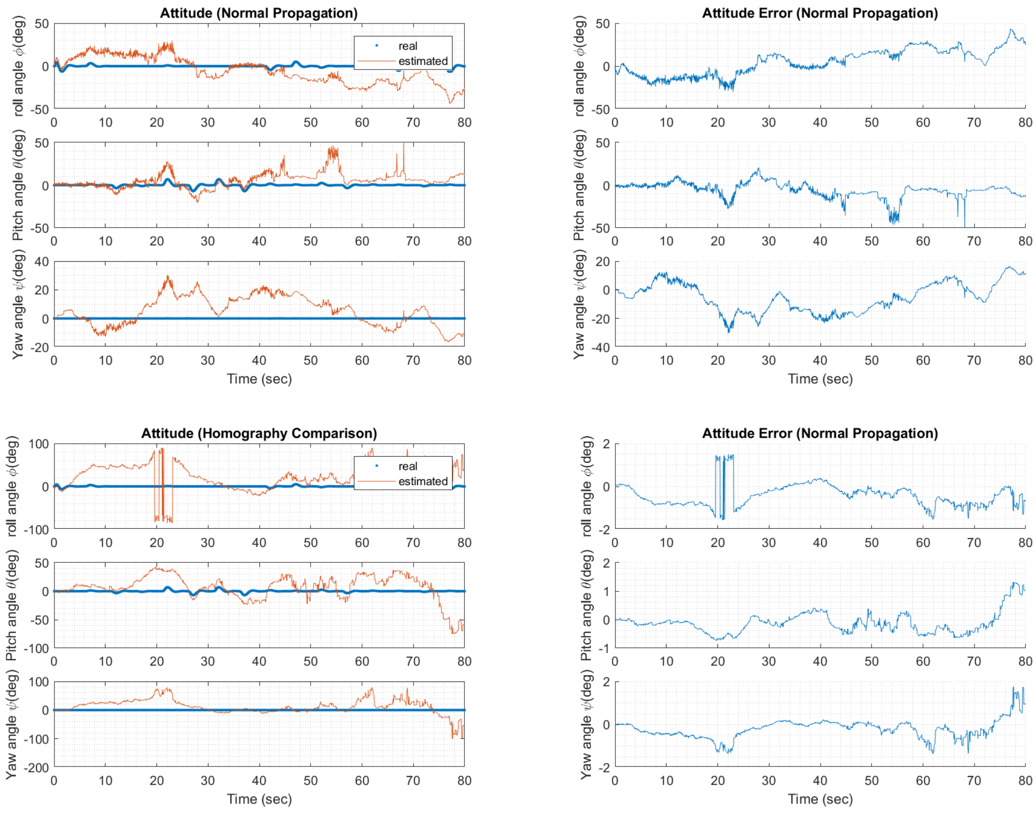


Figure 5.5: Visual Odometry Solution Position Comparison

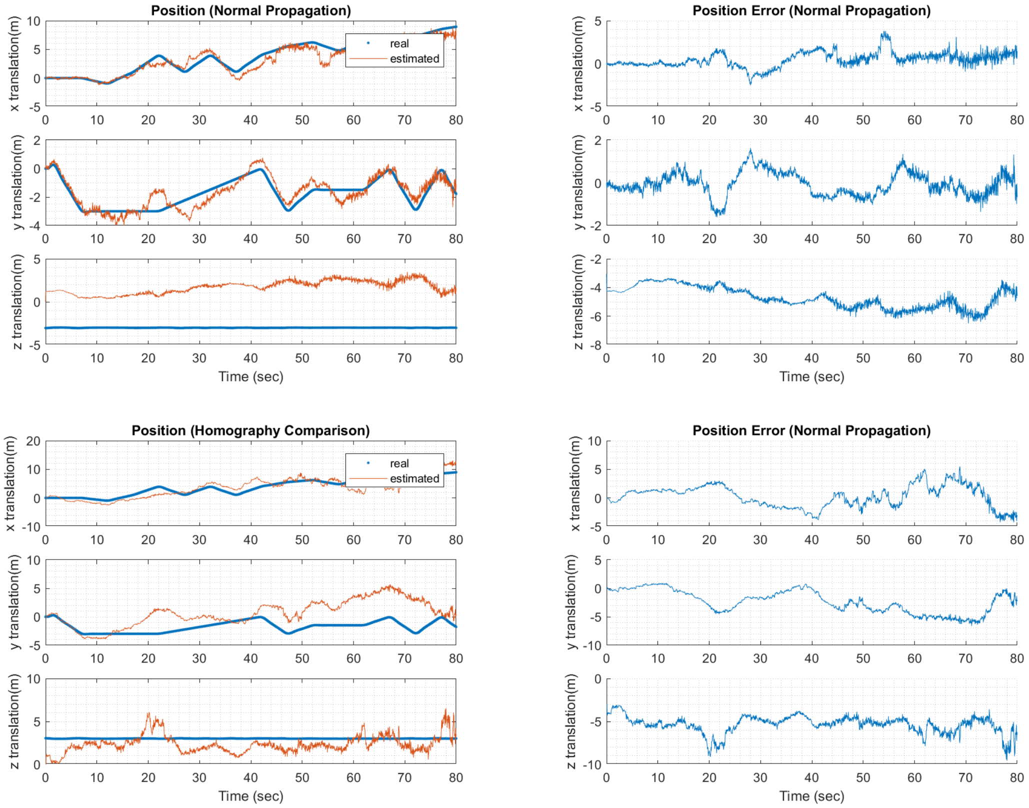


Figure 5.6: Visual Odometry Solution Attitude Comparison

It is possible to see from Figure 5.6 and 5.5 that, despite the generation of a tracking solution, the error accumulates on the estimation due to the daisy-chained approach. Additionally, it is important to remark that the system does not take into account error handling. Therefore, as it can be seen from the corresponding error plots, the error accumulates as a function of time.

The performance analysis of each method is done by computing the root mean square (RMS) error in Table 5.2. It is possible to observe from Table 5.2 that the overall performance using the normal propagation is better than the homography comparison method. This effect occurs because at some instances of time, the homography comparison method could select a false disambiguation since the homography estimation always leads to the computation of different singular values when decomposing the homography in presence of noise. An example of this phenomenon takes place at time $t = 20$ where a possible false lock occurs. However, the overall performance of the algorithm can be improved by adding redundancy to the comparison metric, thus eliminating false locks. This is accomplished by adding a greater number of comparisons at the expense of the increment of the time delay. The same effect can be accomplished by introducing an adaptive tolerance value as suggested by (Caballero et al., 2009). However, the addition of an adaptive tolerance generates an iteration at each time step, thus increasing the computational expense.

Table 5.2: RMS error computation for Visual Odometry sample test

		RMS Error	
State	**Units**	**Normal Propagation**	**Homography Comparison**
X_N	[m]	1.0584	9.0174
Y_N	[m]	1.1221	4.8862
Z_N	[m]	4.7549	8.1633
ϕ	[rad]	0.2340	1.0342
θ	[rad]	0.2697	0.3530
ψ	[rad]	0.4407	1.2819

5.3.2 Numerical Observability Analysis Results

In this section, a numerical test of observability is provided based on the equations and metrics described in Section 5.2.3. Similar to the Visual Odometry Results, a simulation model was provided with exactly four corners on the ground so the exact solution for homography could be computed. The motion dynamics used for the observability analysis were based on the 3D trajectory shown in Figure 5.7. The errors used for the computation of the covariance matrix correspond to the variances detailed in Table 5.1 and correspond to the variances of a low-grade micro-eletromechanical system (MEMS) IMU. The initial covariance matrix P_0 values used on the observability study are listed in Table 5.3.

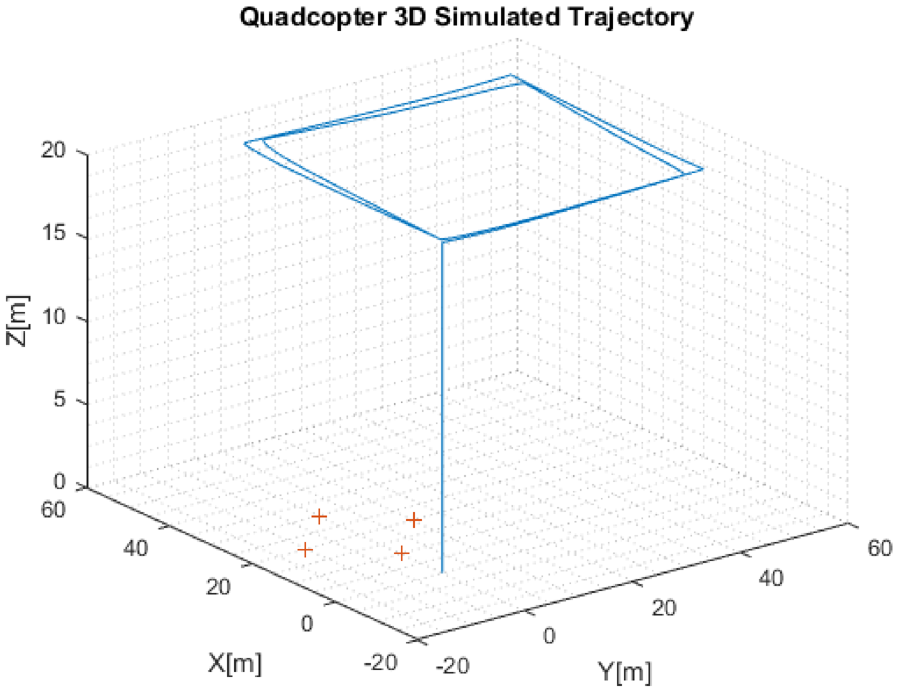


Figure 5.7: 3D Trajectory for Numerical Observability Analysis

Table 5.3: Initial Error Covariance for Observability Analysis

	State	**Covariance (P_0)**
δP^N $[m]$	X_N	1
	Y_N	1
	Z_N	1
δV^N $[m/s]$	V_{XN}	0.5
	V_{YN}	0.5
	V_{ZN}	0.5
$\delta \rho$ $[rad]$	ϕ	0.17
	θ	0.17
	ψ	0.085
Δb_{acc} $[m/s^2]$	b_{ax}	0.01
	b_{ay}	0.01
	b_{az}	0.01
Δb_{gyr} $[rad/s]$	b_p	0.01
	b_q	0.01
	b_r	0.01

The observability analysis was done by computing the discrete observability gramian matrix using Eq 5.5 for the time span of $T = [0, 25]$ sec. The resultant gramian for the aforementioned time span and the system was computed and then analyzed for observability by performing the rank test from Eq. 5.39. Thus, the resultant singular values are shown in descendant order in Table 5.4. As Table 5.4 shows, there exist two states which are considered to be unobservable since the corresponding singular values are zero.

Table 5.4: Singular Values for the Observability Gramian Matrix

Index #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Singular Value	3.20E+08	2.34E+08	8.56E+07	5.81E+07	2.25E+06	1.76E+06	1.25E+06	3.79E+05	2.84E+05	2.66E+03	2.41E+03	2.12E+03	1.82E+02	0	0

The unobservable nullspace spanned by the zero singular values are computed to be as follows:

$$V_{uo} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

These results show that the translational states X_N and Y_N are unobservable. However, an unobservable subsystem can still be detectable, or in other words, the unobservable states can still be estimated by simple propagation. This detectability condition entails that the unobservable states must be naturally stable.

In order to evaluate how observable are the observable states, the estimability analysis is performed by computing the relative rate of change of the standard deviation of the optimal process covariance $P_{0,k}$ on the time span $T = [0,25]$ *sec* as explained in Eq. 5.39. The resultant rate of change is shown in Table 5.5. From Table 5.5 it is also possible to see that in general, all the observable states are completely estimable, with exception of the bias error states since it is shown that the bias error states are less estimable than the other states. Additionally, it can also be determined that b_p is far less estimable than the rest of the states.

5.3.3 Homography-Based Kalman Filter Results

The model used for the simulation of the results of the homography-based EKF was linked to the virtual environment as explained in Chapter 4. The idea was to test the homography, Ransac outlier rejection, and feature detection and matching algorithms that were described in Section 6.2.2. As it can be seen from Figure 5.8, the homography Ransac algorithm detects planar correspondences despite the existence of obstacles on the ground. Based on the computation of the homography using the simulated camera, the estimated solution for the system was generated.

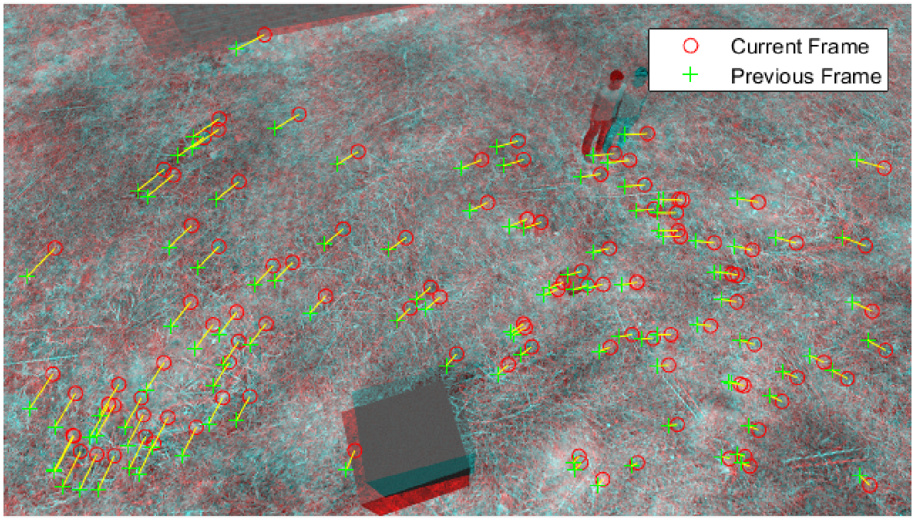


Figure 5.8: Simulated Camera Sample Tracking

Table 5.5: Error Covariance Rate of Change in the Standard Deviation

State	**Covariance Rate**
X_N	0
Y_N	0
Z_N	1
V_{XN}	1
V_{YN}	1
V_{ZN}	1
ϕ	0.9997
θ	1
ψ	1
b_{ax}	0.956
b_{ay}	0.9596
b_{az}	0.9319
b_p	0.67643
b_q	0.992
b_r	0.996

The results for the 3D trajectory comparison are shown in Figure 5.9. Additionally, the estimated state plots are shown in Figure 5.10 where a comparison between the EKF results and a DR solution is shown.

The results generated also include a three sigma error plot on the right column of Figure 5.10. From Figures 5.9 and 5.10 it is possible to see that in general an accurate tracking and estimation is generated where the EKF outperforms the DR solution. However, it is also possible to see that the sigma limits from the error plot in the position in the x and y direction estimation never converge to a constant value. This effect occurs due to the unobservability of the X_N and Y_N states. The unobservability effect on the position states can also be evidenced in the error drift of these states.

Note that the variance in the sigma bounds in the velocity plot contains variations due to the propagation of a DR in between measurement updates; in order words, the uncertainty tends to increase until a measurement corrects the state. Nevertheless, as shown in Table 5.6, when a comparison of the RMS errors of the EKF solution to the DR estimates shows that the general performance of the homography-basd EKF is better than the DR.

Table 5.6: RMS Error Homography EKF Simulation

		RMS Error	
State	**Units**	**DR**	**EKF**
X_N	$[m]$	937.4358	1.1231
Y_N	$[m]$	2245.589	1.6143
Z_N	$[m]$	94.7884	0.0581
V_{XN}	$[m/s]$	23.0335	0.1351
V_{YN}	$[m/s]$	80.3079	0.1151
V_{ZN}	$[m/s]$	2.3790	0.0387
ϕ	$[rad]$	0.1392	0.0034
θ	$[rad]$	0.1554	0.0033
ψ	$[rad]$	12.696	0.0019

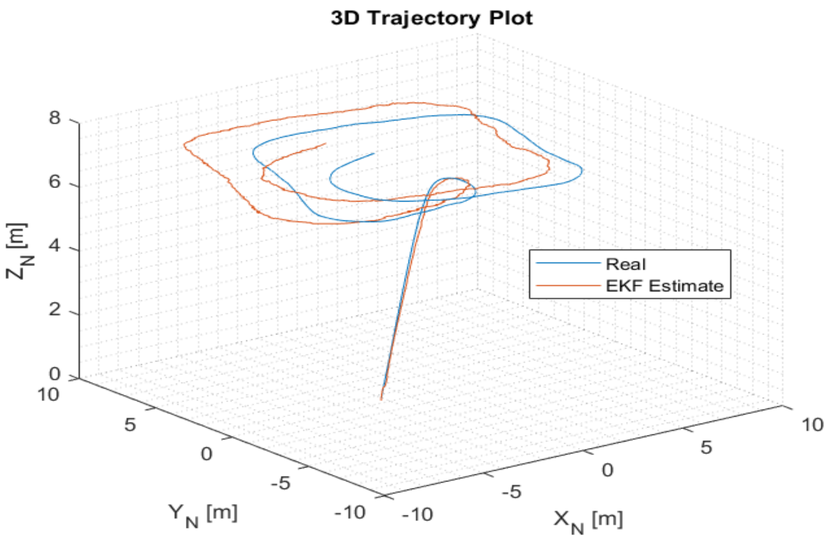
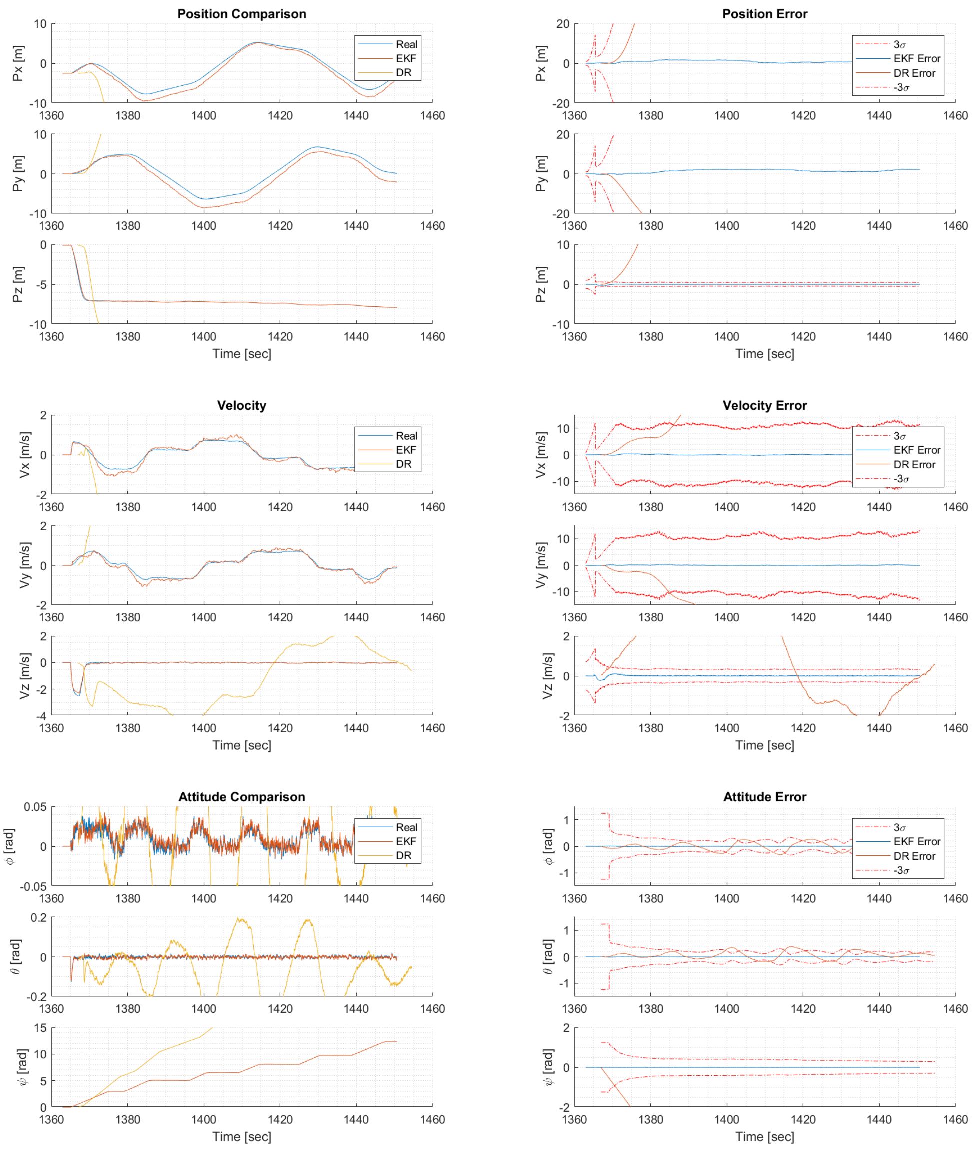


Figure 5.9: Quadcopter Simulated 3D Trajectory for Homography-Based EKF

5.3.4 Monte Carlo Simulation

In order to generate a performance comparison analysis between the visual odometry, dead reckoning, and homography-based EKF, a Monte Carlo simulation was done by varying the noise seeds used in the random number generation for the Gaussian white noise addition. An analysis of 100 runs was made for a system with exactly four feature points. The simulation model given by Hartman et al. (2014) and the trajectory used for the Monte Carlo simulation were the same ones used in the visual odometry simulation shown in Figure 5.9. Note that since the vehicle starts in the air at an altitude h, it was assumed that the first normal information at time $t = 0$ was not available; thus the computation of the disambiguation was done using the *Multiple Homography Comparison* method.


Position Comparison

Position Error

Velocity

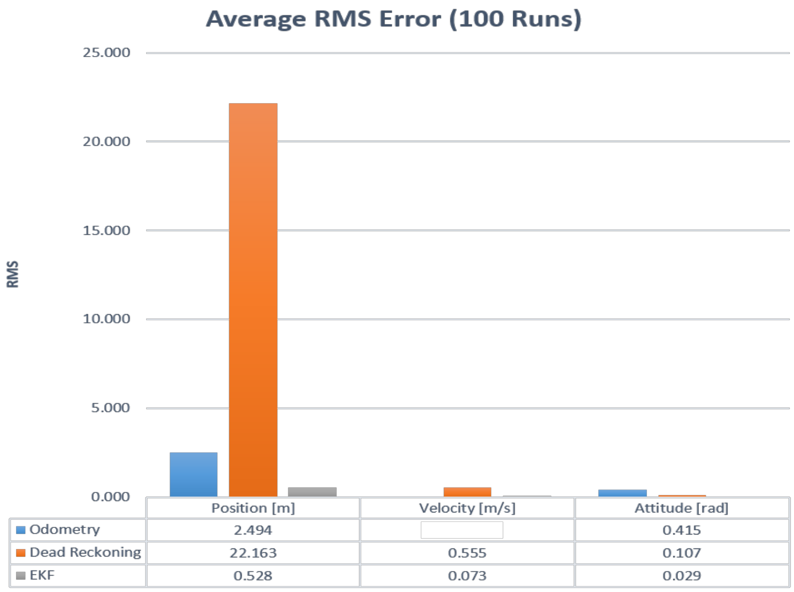
Velocity Error

Attitude Comparison

Attitude Error

Figure 5.10: Homography-Based EKF Simulation Results

The results were averaged for position, velocity, and attitude respectively. A bar graph showing the comparison results for the Monte Carlo simulation is shown in Figure 5.11. It is possible to see that in general the visual odometry provides an estimate that can be more reliable than the dead reckoning solution. However, the visual odometry, due to the lack of a covariance matrix, still accumulates error that drifts as a function of time. Additionally, with the visual odometry solution, there is the lack of a correction for the IMU bias error.



	Position [m]	Velocity [m/s]	Attitude [rad]
🟦 Odometry	2.494		0.415
🟧 Dead Reckoning	22.163	0.555	0.107
⬜ EKF	0.528	0.073	0.029

Figure 5.11: Monte Carlo Simulation Results

6. Experimental Results

In this chapter, a description of the generation of results using experimental data is provided. As part of this discussion, a description of the hardware used for the data collection is given. Additionally, the image feature recognition algorithm is also described together with the handling of false measurements as part of the integration of the homography-based EKF.

6.1 Hardware Description

The testbed used all the data collection in this thesis is a 3DR Iris quadcopter. This is a low-cost testbed that is controlled by the Pixhawk flight controller computer. The main advantage of this testbed is that its stability and control has been largely tested as part of the Ardupilot open source project, making this UAV an ideal testbed for data collection and algorithm development. Additionally, the structure of the Iris quadcopter allows direct attachment of a GoPro camera system. Figure 6.1 shows a close view of the assembled testbed with the instrumentation onboard the vehicle.

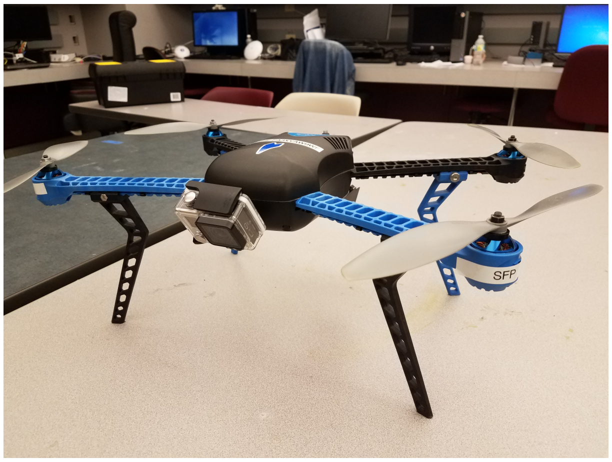


Figure 6.1: 3DR Iris Quadcopter Testbed

The Iris quadcopter specifications are shown in the Table 6.1.

Table 6.1: 3DR Iris Quadcopter specifications

Item	**Description**
Motor to Motor Dimension	550 *mm*
Weight (with battery)	1282 *gr*
3-cell Battery	$11.1\ V,\ 3.5\ Ah$
Motors	AC 2830, 850 *kV*
Telemetry/ Control Frequency	915 *MHz*
Average flight time	10-15 mins
Payload Capacity	425 *gr*

6.1.1 Pixhawk Autopilot

The Pixhawk autopilot is a high-performance autopilot-on-module that was developed as part of an open source and hardware project by The Computer Vision and Geometry Lab and of the Eidgenssische Technische Hochschule (ETH) Zurich. It is a low-cost flight computer suitable for fixed wing, multirotor, cars, and boat testbeds. The autopilot module contains a 168 $MHz/$ 252 $MIPS$ Cortex-M4F processor with a capacity of 256 KB in RAM and 2MB in flash memory. This board is characterized by running an efficient real-time operating system designed with flight control and management in mind. It contains 14 PWM/Servo outputs and several different connectivity options for the inclusion of additional peripherals. The Pixhawk module includes a three-axis ST Micro LSM3030D 14 bits accelerometer/magnetometer that supports standard and fast modes (100 Hz and 400 Hz). Moreover, the Pixhawk autopilot includes an ST Micro L3GD20H 16 bits Gyroscope as a three axis angular rate sensor. As part of the altitude measurement sensor, it uses a MEAS MS5611 barometer. In addition, as means of providing redundancy, a second 3-axis accelerometer/gyroscope is used (Invensense MPU 6000).

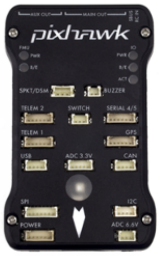


Figure 6.2: Pixhawk Autopilot

6.1.2 3DR GPS Module

This external GPS module provides the integration of a digital compass (HMC5883L) and the GPS receiver (u-Blox LEA-6H). These two sensors run at 3.3 V and provide a measurement update of $5Hz$. Additionally, they are designed to work with the GPS and Galileo constellations.

Figure 6.3: 3DR GPS External Module

6.1.3 GoPro Hero 3+

The camera used for the data collection is the GoPro HERO 3+ that provides wireless connectivity. The images are stored in H.264 digital video format, a proprietary GoPro format, and uses a CMOS optical sensor. The video can be generated at a rate of up to 60 fps with an effective sensor resolution of 5.0 MP. The main characteristic of this camera is that it uses a wide-angle HD lens (fisheye lens) and has a very high impact resistance due to its enclosure.

Figure 6.4: GoPro HERO 3+

6.2 Vision Processing Algorithm

In this section, a description of the process of computing the homography matrix is provided. This includes the calibration and undistortion of the images together with the feature extraction and matching. The overall vision processing workflow is shown in Figure 6.5 where the vision system is divided into separate components that lead to the computation of the Homography matrix that is used in the Homography-based EKF.

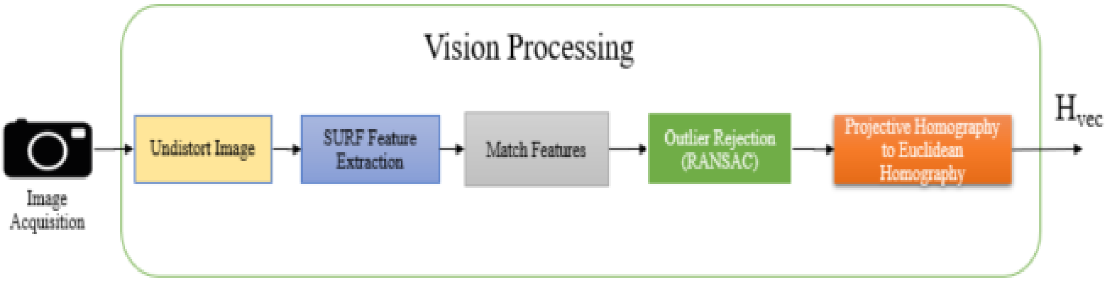


Figure 6.5: Vision Processing Workflow Diagram

6.2.1 Camera Calibration

Camera calibration is the process of the estimation of the camera parameters that form the camera intrinsic matrix, camera extrinsic matrix, and the image distortion coefficients. This is a process usually done before the image feature extraction and uses images of a checkerboard with known distances. In order to generate the camera calibration, the MATLAB single calibration app was used. The typical workflow for the generation of the camera matrix is shown in Figure 6.6.

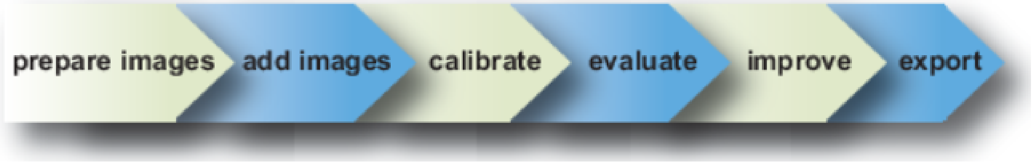


Figure 6.6: Camera Calibration Workflow (Mathworks, n.d.)

As part of the calibration process, a set of 28 photos of a checkerboard was taken from different angles. However, this set of images was pre-undistorted from the fisheye distortion typical from the wide-angle lenses of the GoPro cameras. This process was done using the GoPro Studio app. Once the pre-undistortion was completed, the set of images were input to the MATLAB single camera calibrator app. A sample of the pre-undistorted checkerboard images taken at two different angles is shown in Figure 6.7.

The reprojection of the tracked checkerboard corners in the camera calibration is shown in Figure 6.8 together with the estimated extrinsic parameters for the calibration session.

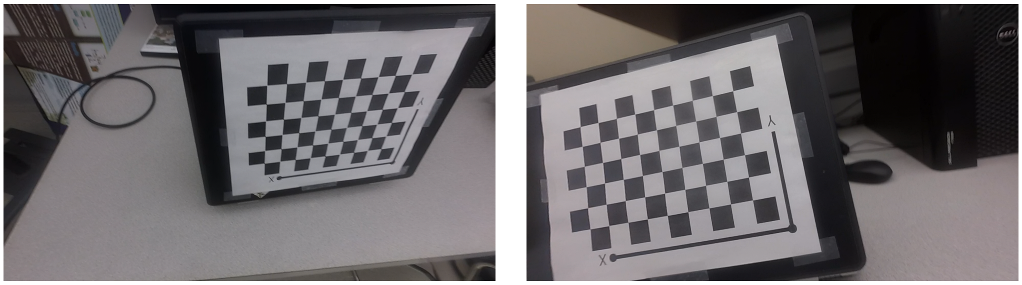


Figure 6.7: Pre-Undistorted Checkerboard Sample Images

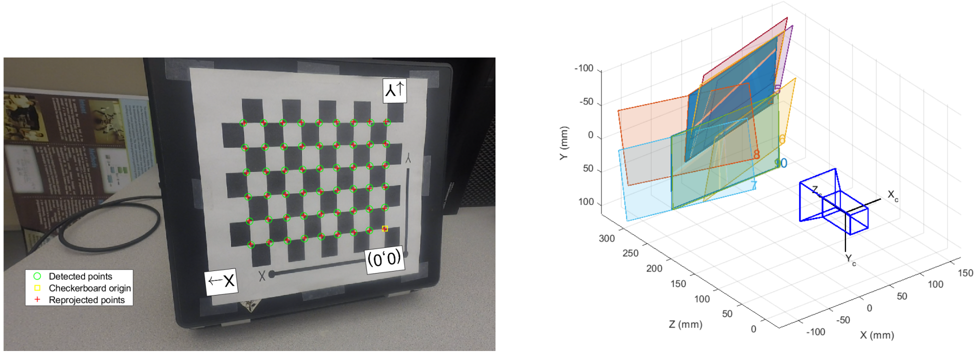


Figure 6.8: Camera calibration Sample Results

Finally, the estimated intrinsic matrix K that will be used to map the homography from the projective to the Euclidean space is:

$$K_{int} = \begin{bmatrix} 840.0015 & 0 & 945.439 \\ 0 & 841.347 & 531.085 \\ 0 & 0 & 1 \end{bmatrix}$$

6.2.2 Image Feature Detection

Once the image from the GoPro camera has been extracted, undistorted, and calibrated, it is necessary to detect ground features that can be matched between frames. For this purpose, the chosen algorithm is the Speeded-Up Robust Features (SURF) due to its robustness, portability, and sensitivity. This algorithm was developed by Bay et al. (2008) as an optimal iteration of the Scale-Invariant Feature Transform (SIFT) algorithm.

The idea behind SURF is to generate an approximation of the Laplacian of Gaussian with a box filter or square-shaped filter for finding scale-space. The advantage of using the approximation is that the convolution with a box filter can be computed faster if the integral image is used. In order to generate a blob detector, SURF relies on the determinant of the Hessian matrix to compute the scale and location of the points of interest in the image as shown in Figure 6.9 (Bradski, 2000). Therefore, given a pixel point $p = (x, y)$, the Hessian matrix $H(p, \sigma)$ at point p and scale σ is given by:

$$H(p,\sigma) = \begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix} \tag{6.1}$$

given that L_{mn} represent the convolution of the second-order derivative of the Gaussian of the image (Bay et al., 2008).

Additionally, the feature description in SURF is based on wavelet responses in vertical and horizontal directions. This description is taken around a neighborhood of size 20×20 squares divided in 4×4 subregions. The vector descriptor takes the form of

$$\nu = (\sum d_x,\ \sum d_y,\ \sum |d_x|, \sum |dy|)$$

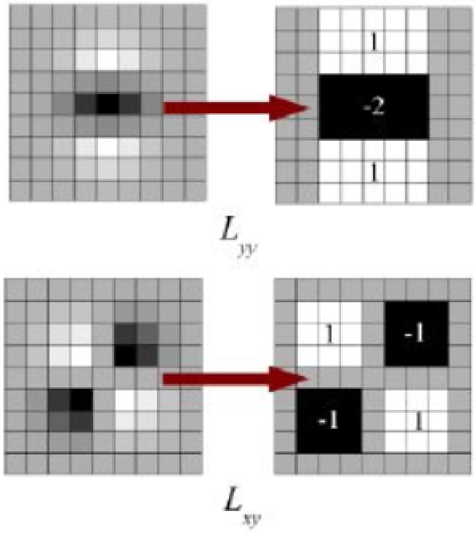


Figure 6.9: Hessian of an Image for SURF Feature Computation (Bay et al., 2008)

making the descriptor vector ν of size $R^{64\times1}$ (Bradski, 2000).

Once the feature has been detected in the current and previous frame, it is necessary to generate a feature index pair correspondence so the Homography matrix can be computed. The simplest, yet computationally expensive, matching algorithm can be defined as a Brute-Force Matcher where a descriptor of a feature in the first set of features is matched with all other features in the second set using some distance calculation, returning the closest one as the correspondence pair. A sample of the SURF feature detection in two subsequent frames is shown in Figure 6.10 where the matching correspondences in the two frames are represented with a connecting yellow line.

From Figure 6.10 it is possible to see that the algorithm was able to find a set of matching correspondences. However, it is also possible to see that there exist several lines that are not parallel, but rather inclined in several directions. This means that several false matches

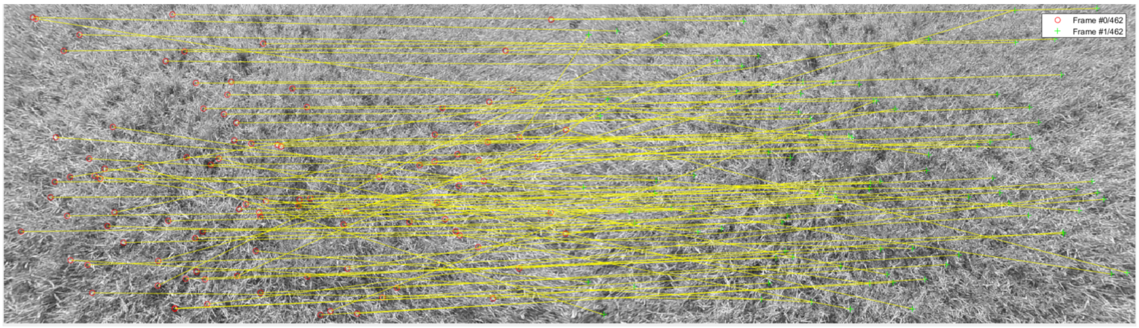


Figure 6.10: SURF Feature Extraction Sample Dataset

were generated; these are called outliers. Therefore, in order to generate an accurate feature matching, an outlier rejection algorithm is required. The outlier rejection algorithm used was RANSAC and is described in Algorithm 3.1. The output of the RANSAC algorithm is the set of feature inliers and the projective homography matrix. A sample of the matching after RANSAC is shown in Figure 6.11.

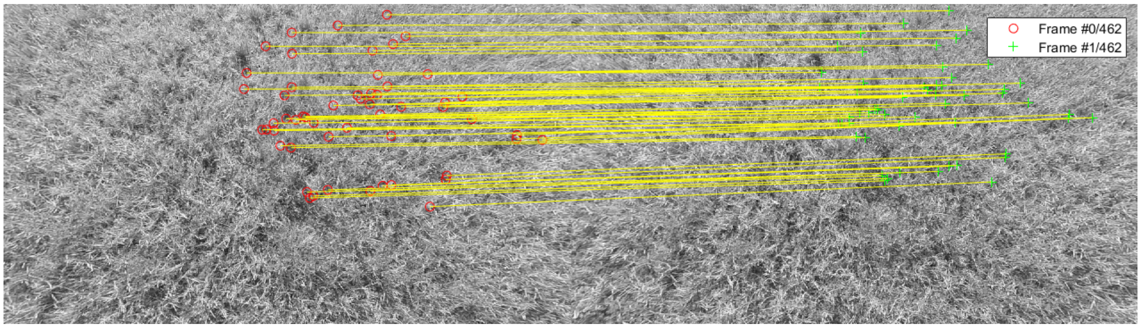


Figure 6.11: SURF Features with Outlier Rejection Sample

6.3 Measurement Integration

Despite the implementation of outlier rejection algorithms such as RANSAC, there exists the possibility that the computed homography matrix contains a reliability of less than 90%, resulting in a wrong projective transformation. This means that the RANSAC algorithm was not able to filter all of the outliers in the data set. Specifically, this effect is mostly seen when the vehicle is close to the ground at takeoff and landing stages due to the rapid change of scale of the SURF features in the image plane. A sample of each of the nine parameters of the homography measurement history of a flight test is shown in Figure 6.12. Here, it is possible to observe that there exist peaks that are out of magnitude at the beginning and the end of the flight, portions of the flight that corresponds to the takeoff and landing respectively.

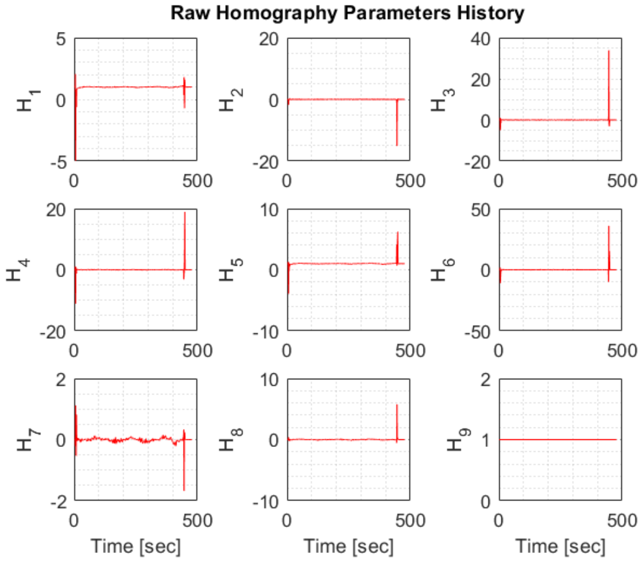


Figure 6.12: Raw Homography Measurement History

In order to solve this problem, there exist several different methods that can be taken into account. One solution is to start the vision state estimation when the vehicle is in loitering mode at a slowly changing altitude rate. However, the false inliers can still occur if the vehicle generates sudden rapid moves, thus generating false measurements that can contribute to a faster drift. Therefore, two filtering methods are considered as part of the scope of this thesis. The first one is a digital median filter, and the second one is an innovation filter. A complete outline of the integration of these two methods is shown in Figure 6.13.

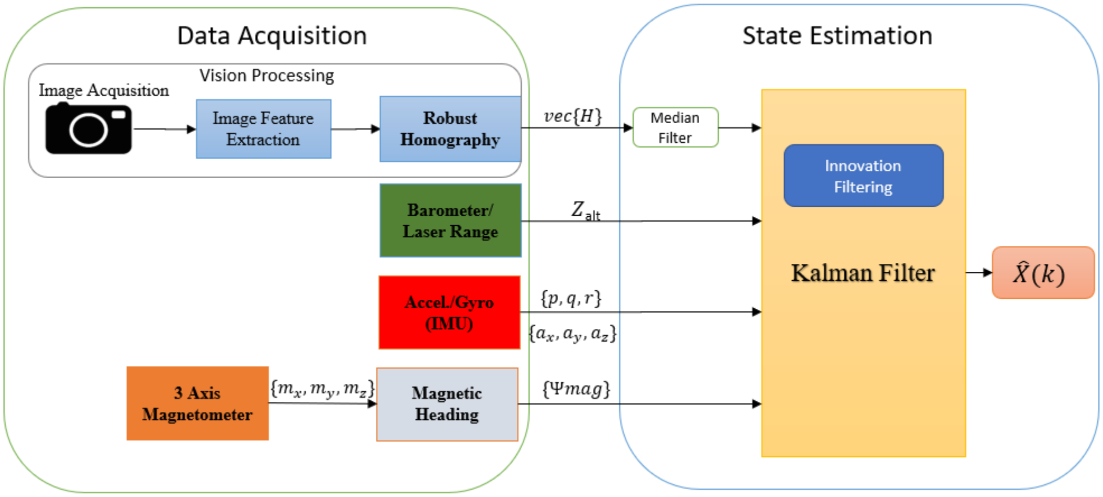


Figure 6.13: EKF Diagram with Measurement Outlier Rejection

6.3.1 Median Filtering (MF)

The median filtering is a nonlinear digital method commonly used for noise removal from images while preserving edges. Therefore, it is usually used in imaging as a ”salt and pepper” type nose removal. It is similar to a mean filter. However, the characteristic of this filter is that it preserves useful data detail in an image. This type of filtering is particularly useful when the distribution function that characterizes the noise is unknown.

Typically, it can be assumed that the homography matrix contains a Gaussian distribution error noise, but when false inliers appear, the distribution of the error function changes. Therefore, a reasonable approach is to assume that the density is a member of the measurement set or some family of parametric families (Hamza et al., 2001).

Consider the additive noise model where S_i is a discrete *m-dimensional* deterministic sequence corrupted by a zero-mean noise sequence V_i with unknown distribution function, where index $i \in \mathbb{Z}^m$.

$$X_i = S_i + V_i \tag{6.2}$$

where X_i is the observed sequence. The objective is to generate an output measurement Y_i as a function of the raw measurement X_i; thus $Z_i = f(X_i)$ where f represents the filtering operator. Then, let W be defined as a sliding window subset of $\mathbb{Z}^m$ of size $2N+1$ and W_i the window data sequence centered at the i location. Then the sliding windows can be defined as

$$W_i = \{X_{i+r} : r \in W\}$$

Applying the operator function *med* to the sliding window Wi, the output of the median filter takes the following form:

$$Y_i = med\{W_i\} = med\{X_{i+r} : r \in W\} \tag{6.3}$$

(Hamza et al., 1999)

When applying Eq. 6.3 with a sliding window of size 3 to the homography measurement, the magnitude of the false inliers peaks can be reduced as shown in Figure 6.14. This can help with the reduction of uncertainty when the covariance in the homography-based EKF has not converged yet. However, when the magnitude of the median filter is too high, there exist the risk of filtering useful information contained in the homography matrix. Therefore, after some experimentation, it was found that the optimal median filtering magnitude for the purposes of the Homography EKF is of size 2.

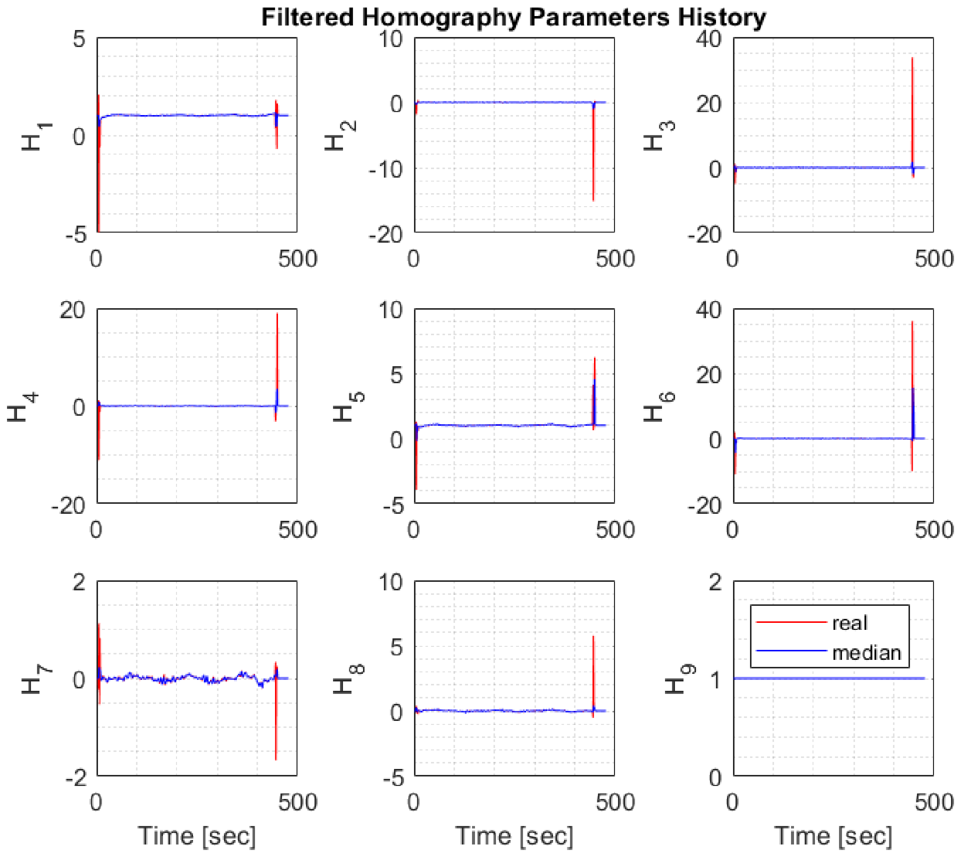


Figure 6.14: Homography Parameters History with Median Filtering

6.3.2 Innovation Filtering (IF)

Another type of filtering considered for the correction of the false inlier measurements is the idea of using an innovation filter or more commonly known as measurement gating or editing. The idea is to use the innovation measurements from the Kalman filter to determine whether the incoming measurement updates are consistent with previous information. This is done by normalizing the measurement and comparing it with a predefined threshold value. Once the false information has been determined, it is possible to modify the Kalman gain so the filter can take into account the false measurement by rejecting the corresponding rows of the measurement Jacobian H_k and the corresponding rows of the measurement covariance matrix R.

Given a homography measurement subset $H_{vec\,i} \in \mathbb{Z}^m$ of the first and third term of the homography parameters as plotted in Figure 6.15, it is possible to see that there exist peaks in the innovation δZ^- that correspond to a false measurement as seen in the innovation covariance plot on the right side of each parameter. However, the magnitude of the false inlier in the innovation of the first homography parameter H_{vec1} differs from the magnitude of the third homography parameter H_{vec3}. Therefore, it is necessary to generate a normalization of the innovation so the innovations have a zero-mean unit-variance Gaussian distribution and that the innovations covariance assumed by the Kalman filter is true (Groves, 2013).

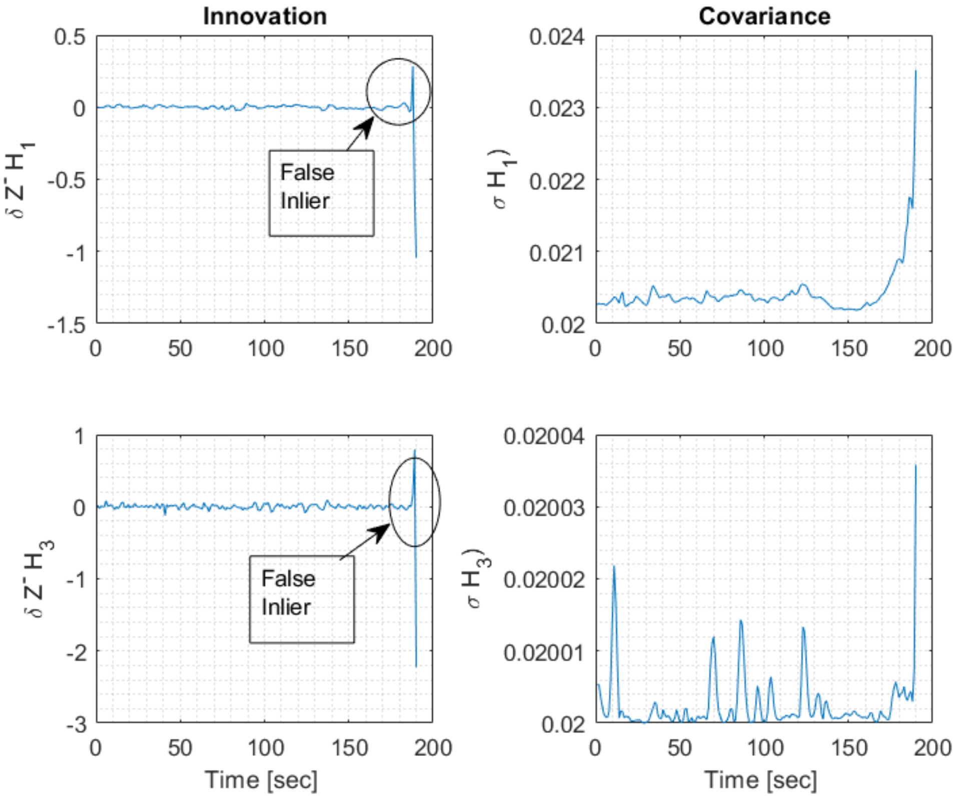


Figure 6.15: Measurement Innovations with Kalman Innovation Covariance

The innovations are thus normalized given the following equations derived from Eq. **??** and Eq. 5.35.

$$\delta Z_k^- = Z_k - h(\hat{x}_k^-) \tag{6.4}$$

$$C_{\delta Z_k^-} = H_k P_k^- H_k^T + R_k \tag{6.5}$$

where the normalized innovation is :

$$y_{k,j}^- = \frac{\delta Z_{k,j}^-}{\sqrt{C_{\delta Z_k^-}}} \tag{6.6}$$

Figure 6.16 shows the normalized innovations of the homography parameters H_{vec1} and H_{vec3}.

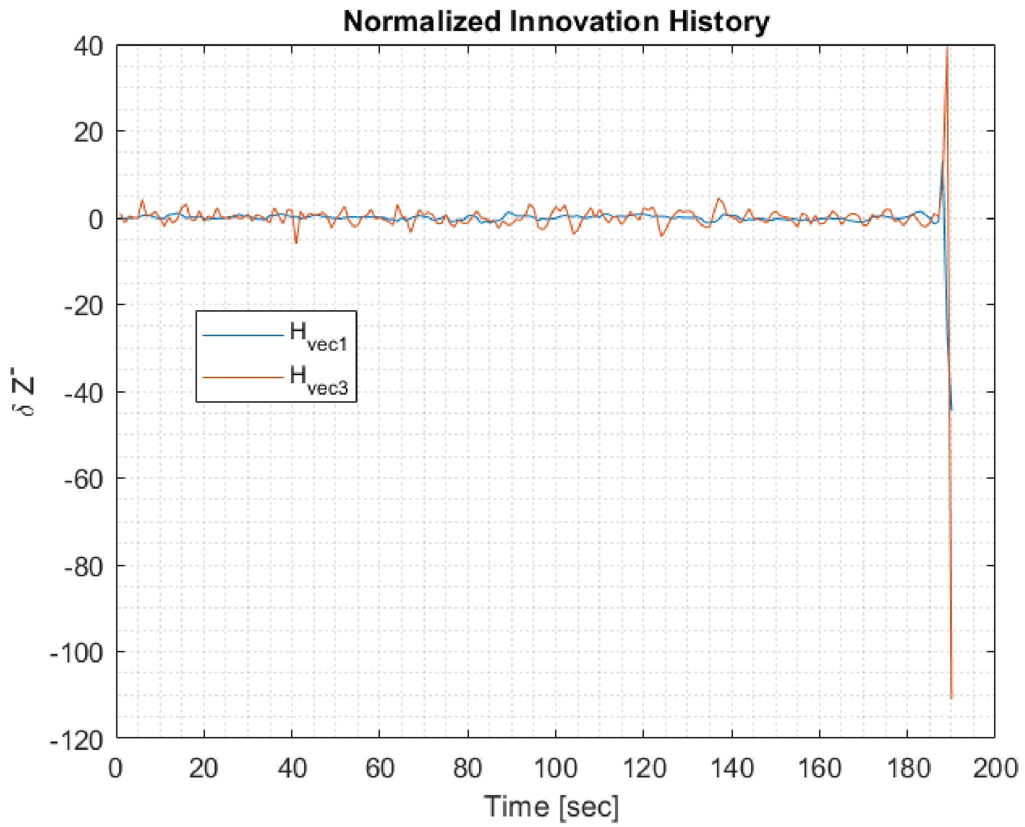


Figure 6.16: Normalized Innovation Homography History

From Figure 6.16 it is possible to see that the magnitudes of the innovations are similar. This allows the definition of a common threshold that would filter the false measurements. However, the relationship between the threshold and the false alarm rate are variant, thus making the assessment of the threshold empirical. Nevertheless, some studies show that a threshold of ±3 allows 99.73% of the genuine measurements to pass. The result of the innovation filtering can be evidenced in the reconstruction of the homography measurement using Eqs. 5.22 → 5.27 as shown in Figure 6.17 where the false measurement peaks in the homography do not influence the reconstruction of the measurement.

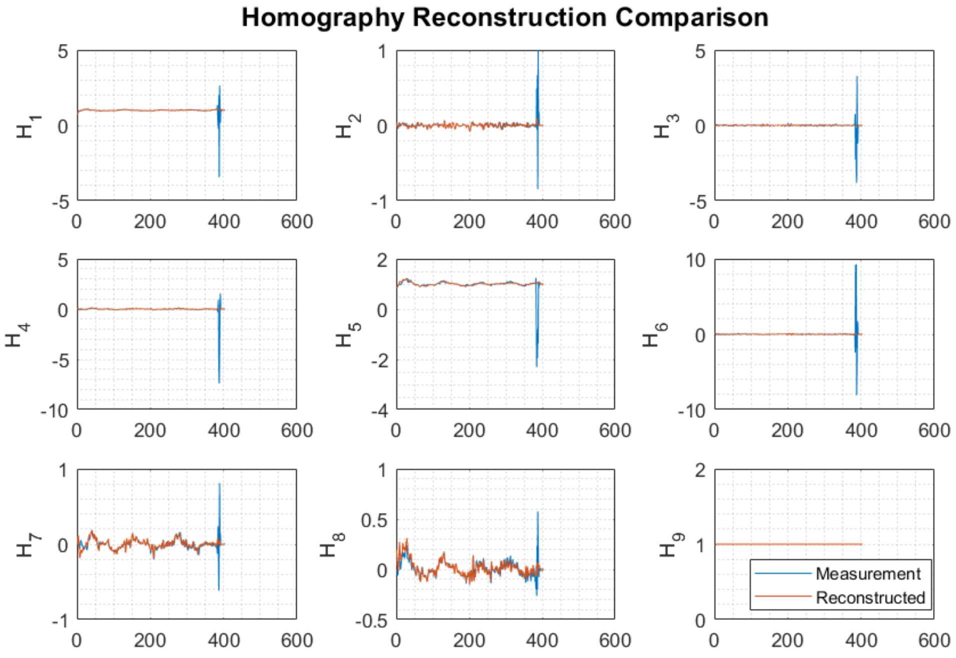
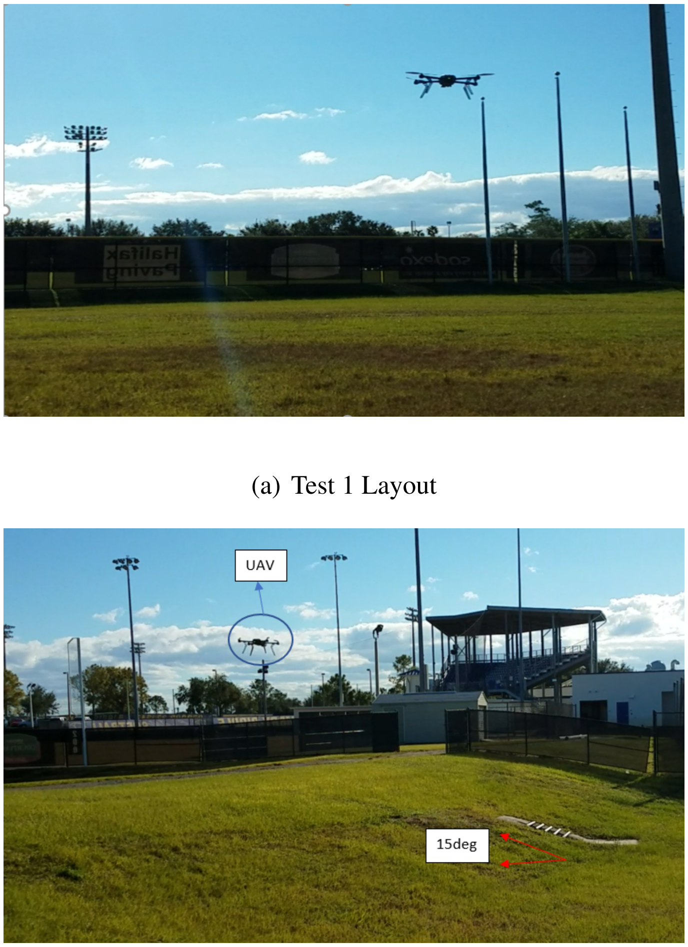


Figure 6.17: Homography Reconstruction Comparison

6.4 Results

In order to test the algorithm, two flight tests were designed. The first one was designed with the requirement of flight over a planar surface with distinguishable ground features that can be detected by the vision system as shown in Figure 6.18(a). The second flight test was designed to be flown over an inclined plane of approximately 15 deg with planar features. This flight test was designed to test the normal vector assumption made in Eq. 5.24. A graphical description of test two is shown in Figure 6.18(b). The two flights were flown in a circular pattern as shown in Figure 6.19 where the GPS trajectory was plotted on top of a Google Earth map. In both flights, the camera was mounted at an angle of +30 deg with respect to the horizontal plane of the vehicle.



(a) Test 1 Layout

(b) Test 2 Layout

Figure 6.18: Mission Tests Description

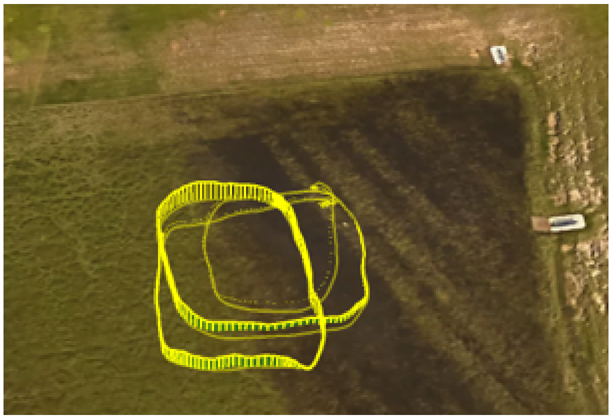


Figure 6.19: 3D GPS-Based Trajectory

The two sets of results were compared against the GPS-based Kalman filter of the pixhawk log and the dead reckoning solution. Additionally, the GPS-based EKF solution was assumed to be correct and accurate, thus a computation of the RMS error of the homography-based EKF and dead reckoning for the two flight tests were computed with respect to the GPS solution.

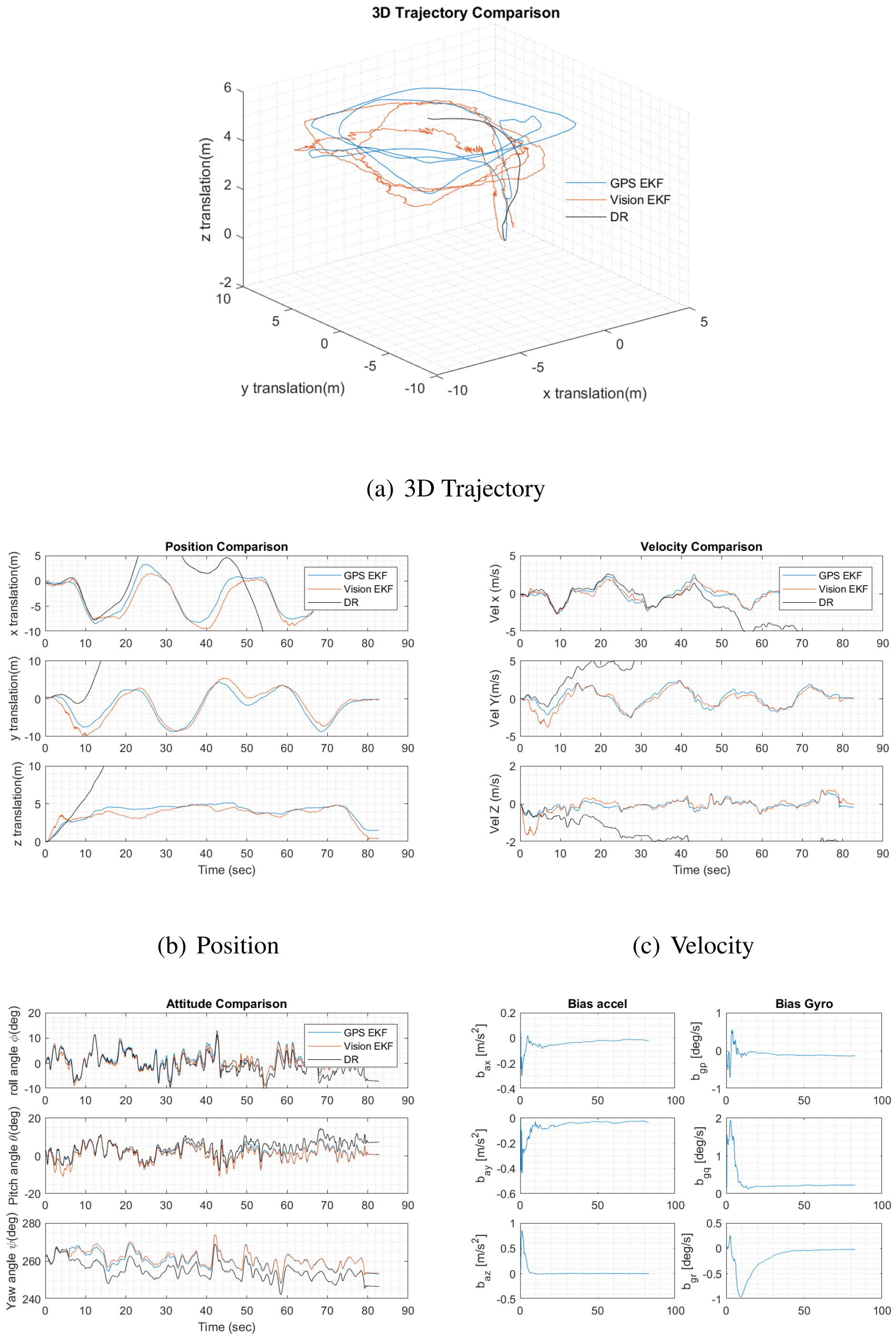
6.4.1 Test 1

The results generated for flight test 1 were computed using just the innovation filter as means of false measurement filtering since the computation of false measurements was sufficient to generate the desired accuracy. From the results shown in Figure 6.20 it is possible to see that the dead reckoning solution tends to generate a fast drift that deviates the solution from the considered true solution. However, the vision-based EKF solution generates a solution that is close to the GPS-based solution despite the false inliers generated at takeoff and landing. However, it is possible to observe in Figure 6.20(b) that the position solution in the X_N and Y_N axis tends to drift away due to the relativistic nature of the homography matrix. Nevertheless, the solution in position, despite being unobservable, generates a slower drift than the one generated by the dead reckoning solution, as predicted in the simulation.

In addition, it is possible to see that the biases in the accelerometers and gyroscopes converge to static values respectively. The RMS error performance comparison of the system is shown in Table 6.2.

Table 6.2: RMS Error Test 1

		RMS Error	
State	**Units**	**DR**	**EKF+IF**
X_N	$[m]$	55.8503	1.1365
Y_N	$[m]$	574.559	1.3454
Z_N	$[m]$	73.8184	0.5941
V_{XN}	$[m/s]$	4.0583	0.3188
V_{YN}	$[m/s]$	24.66	0.5344
V_{ZN}	$[m/s]$	2.0217	0.2296
ϕ	$[rad]$	0.0477	0.0154
θ	$[rad]$	0.0587	0.0223
ψ	$[rad]$	0.1018	0.0175



(a) 3D Trajectory

(b) Position

(c) Velocity

(d) Attitude

(e) Bias

Figure 6.20: Results Flight Test 1

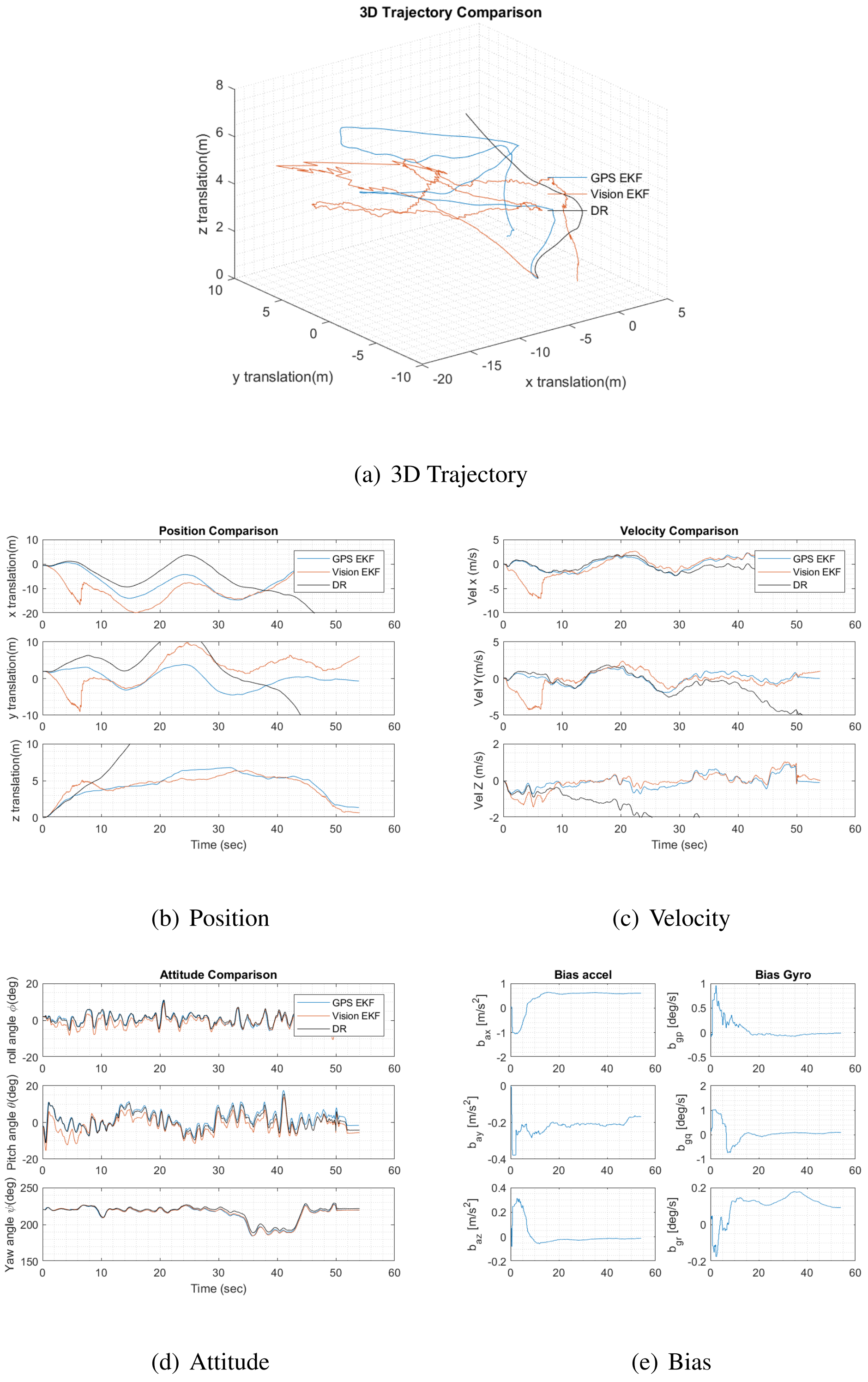
6.4.2 Test 2

Similar to the previous case, the RMS error performance comparison analysis was generated for flight test 2. However, for the generation of results in this flight test, it was necessary to include a median filter since the Ransac homography computation system generated several sets of false inliers not only during takeoff and landing, but also during loitering due to the slanted ground.

As it can be seen from Figure 6.21, the proposed vision-based EKF still generates a better approximation than the DR solution. However, the drift is greater than the drift generated during flight test 1. This is because the addition of an inclined plane breaks the assumption made in Eq. 5.24 and Eq. 5.25 since the approximations of a horizontal planar surface and a vertical absolute distance for all tracked points no longer hold. A possible solution to this problem would be to include a stereo vision system that would allow the computation of the distance of each tracked point relative to the vehicle. However, it is also possible to assume that all features belong to a horizontal planar space if the UAV altitude is sufficient enough to discard small variations in the altitude of the feature points. Nevertheless, despite the broken assumptions, the homography-based EKF still generates a close approximate solution with the usage of the median and innovation filters.

Table 6.3: RMS Error Test 2

		RMS Error	
State	**Units**	**DR**	**EKF+IF+MF**
X_N	$[m]$	12.3358	4.5905
Y_N	$[m]$	15.0471	4.9683
Z_N	$[m]$	42.1822	0.7204
V_{XN}	$[m/s]$	1.4548	1.6177
V_{YN}	$[m/s]$	2.1646	1.3277
V_{ZN}	$[m/s]$	2.1076	0.2652
ϕ	$[rad]$	0.0196	0.0364
θ	$[rad]$	0.0313	0.0682
ψ	$[rad]$	0.0403	0.0163



(a) 3D Trajectory

(b) Position

(c) Velocity

(d) Attitude

(e) Bias

Figure 6.21: Results Flight Test 2

7. Hybrid State Estimation

From the results generated either from simulation or experiments, the homography-based EKF generates a close to accurate estimation of all of the vehicle states. However, the estimated position, despite generating a similar solution to a GPS-based filter, is not observable and tends to drift as a function of time. A slowly paced drift can still generate problems related to stability when the control loop of the vehicle is closed. Therefore, it is desired to reset the error in the position with the integration of a measurement that makes the position states X_n observable without compromising the stability or computational requirements of the current system. For this purpose, an additional measurement derived in Myhre et al. (2018), which is based on the knowledge of the location of known landmarks, was used to correct the state estimates when the landmarks are in the field of view of the camera. In real life scenarios, the navigation reference frame is centered with respect to the landmarks which can take the form of any terrain characteristics at beginning of the filter.

The proposed mission for the exploration of a GPS denied environment is shown in Figure 7.1, where a predefined landing target is located at the starting point of the exploration and is also set as the center of the navigation-reference frame. The vehicle follows a path that obeys a circular pattern so the vehicle explores the terrain and then returns to the starting point. The algorithm is thus divided into three stages. The first stage is called *Landmark Measurement* in which the landing target is in the field of view of the camera. The second stage is the *Homography Measurement* in which the system generates an estimate that relies only on the homography measurement and the inertial sensors information as explained in Section 5.2. The last stage of the navigation system is called *Estimation Correction* in which the landmark measurements are fused with the homography measurements to bound the uncertainty and reset the error drift.

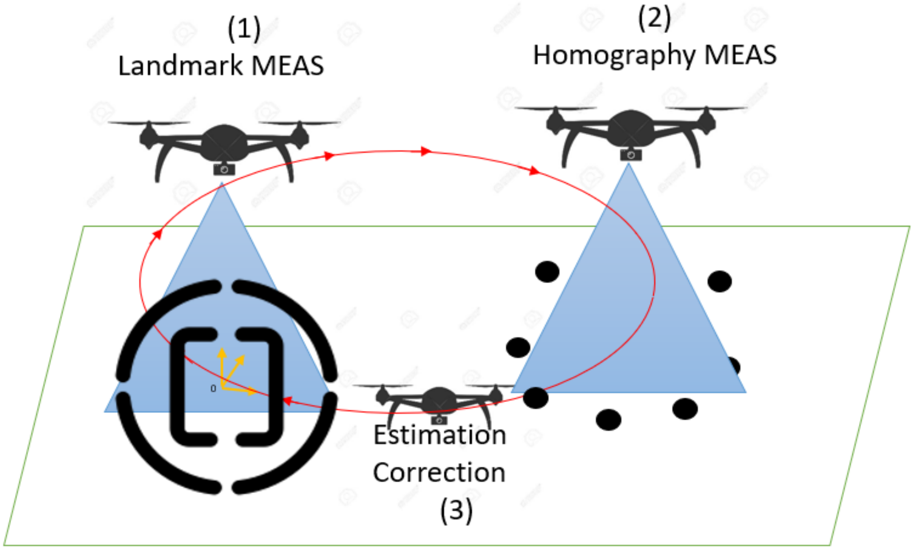


Figure 7.1: Hybrid State Estimation Mission Layout and Description

7.1 Landmark-Based Measurement

The landmark-based measurement is based on the prediction and comparison of the projected landmarks using the pinhole camera model described in Section 4.2.2. Let $\vec{Y}_i$ be the i-th landmark location in the image pixel frame obtained from a vision processing algorithm, such as Viola-Jones or color-based detection. Thus the projected landmark model takes the following form:

$$\hat{Y}_i = K_{int} R_C^N(k)[I|\hat{P}^N(k)]\hat{X}_{lnd,i}^N \tag{7.1}$$

where $\hat{X}_{lnd,i}^N$ denotes the known coordinate location of the i-th landmark in the navigation reference frame, K_{int} represents the intrinsic matrix computed by the calibration matrix procedure explained in Section 6.2.1, $R_C^N(k)$ represents the rotation matrix from camera to navigation frame at time k, $\hat{P}^N(k)$ represents the location of vehicle at time k, and $\hat{Y}_i = \{x_{cam}, y_{cam}, 1\}$ represents the projected landmark on the focal plane in homogeneous pixel coordinates. Eq. 7.1 is applied to every point member of the known landing target $\vec{Y}_i \in \mathbb{R}^n$. The complete landmark measurement vector $\vec{Z}_{lnd,k}$ is of size n and is given as:

$$\vec{Z}_{lnd,k} = \begin{Bmatrix} \vec{Y}_1 \\ \vdots \\ \vec{Y}_n \end{Bmatrix} \tag{7.2}$$

Since not all the measurement landmarks are always inside the field of view of the camera, a logical vector $\vec{\Upsilon} \in \mathbb{R}^n$ is provided as a part of the landmark vision system where $\Upsilon_i = \{1|0\}$ can take the form of 1 if the i-th landmark is present in the camera or 0 if it is not currently present. As a result, the size of vector $\vec{Z}_{lnd,k}$ is $m \leq n$, where m is the number of landmark points that appear on the image.

7.2 Navigation Filter Architecture

The main problem with the integration of the two different vision-based systems in a cascaded manner is that the landmark-based navigation EKf solution has a problem of instability when the landmarks leave the field of view of the camera, thus creating what is known as *black box* navigation. In the black box navigation, the system does not have any means to estimate and correct the covariance of the process, thus causing a divergence in the uncertainty estimation. Additionally, when fusing a black box navigation in cascade, there is a problem of propagation of the covariance error throughout all of the independent systems. A cascaded measurement EKF filter integration was considered in Chavez Armijos et al. (2017). However, for a centralized EKF with cascaded measurements, despite having the capacity of handling black box navigation, the complexity increases due to the increment of the dimension in the measurement model (Groves, 2013). An increment of the dimension and non-linearities in the measurement models also increases the difficulty for measurement fault detection and rejection, which are essential for the homography-based measurement. Therefore, a Federated Filtered Integration (FFI), proposed by Carlson (1990), was considered so the homography-based and landmark-based EKFs could be able to run independently. For this purpose, the information fusion algorithm chosen is a *Single Epoch* (SE) algorithm, which is based on a least squares estimation (Gong & Zhang, 2016; Groves, 2013). The information fusion is given as follows:

$$\hat{X}_k, se = \left(H^T \; P_f^{-1} \; H\right)^{-1} \; H^T * P_f^{-1} \vec{X}_f \tag{7.3}$$

where $\hat{X}_k, se$ denotes the fused SE estimation at time k and $\vec{X}_f = \{\hat{x}_{k,lnd}, \hat{x}_{k,hom}\}$ is a concatenated vector containing the landmark EKF estimate ($\hat{x}_{k,lnd}$) and the homography EKF estimate ($\hat{x}_{k,hom}$). Similarly, $P_F = blckdiag\{\hat{P}_{k,lnd}, \hat{P}_{k,hom}\}$ is a block diagonal matrix concatenation of the last landmark EKF estimate covariance ($\hat{P}_{k,lnd}$) and the last homography

EKF estimate covariance ($\hat{P}_{k,hom}$). Additionally, $H = [I_{15\times15}, I_{15\times15}]^T$ represents a matrix column concatenation of identity matrices. The covariance of the single epoch fusion solution is as follows:

$$P_{k,se} = \left(H^T \, P_f^{-1} \, H\right)^{-1} \tag{7.4}$$

The cascaded architecture of the FFI with single epoch information fusion is shown in Figure 7.2.

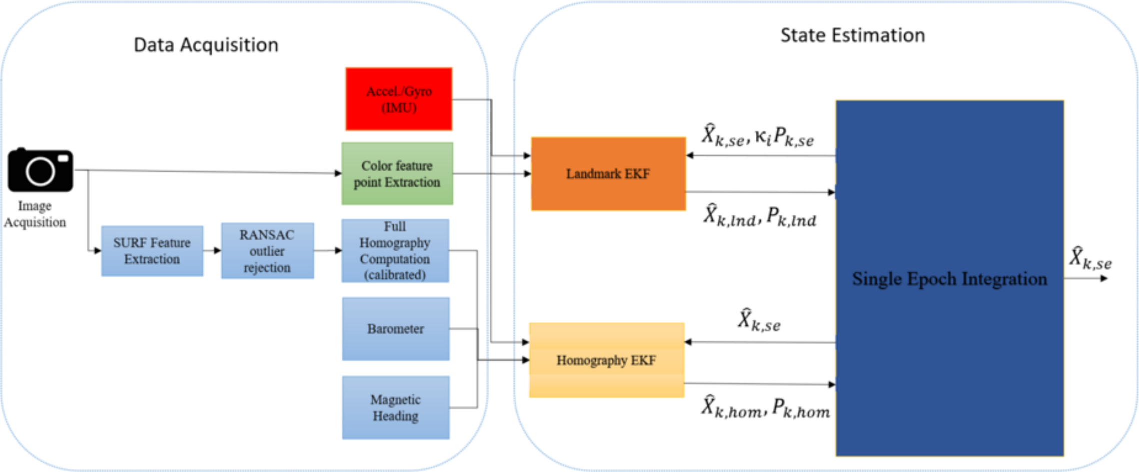


Figure 7.2: Federated Hybrid Filtered Integration

As it can be seen from Figure 7.2, the two subfilters use the previous estimate of the centralized estimation. However, it is possible to see from the definition of P_f and Eq. 7.4 that the SE algorithm defines the covariance matrix with an error correlation between the two different subsystems. This correlation can affect the computation and performance of the homography EKF estimation covariance if the landmark EKF enters into black box navigation. Therefore, the fused covariance $P_{k,se}$ is multiplied by a constant $\kappa_i = 3$ before the covariance is fed back into the landmark EKF so a fusion reset can be computed. A reset on the fusion is done by underestimating the fused $P_{k,se}$ from the single epoch integration, meaning that it is multiplied by a constant. For the homography EKF, there exists no reset on the information, meaning that the last estimation of the SE covariance is not fed back into the homography EKF.

7.3 Experimental Results

The hybrid filter architecture was tested given a circular trajectory as shown in Figure 7.1 so the three stages of the filter could be tested. The landing target used was based on an ”x” shaped form with balls of different colors at the corners as shown in Figure 7.3. The algorithm used for the landmark detection was extracted from Myhre et al. (2018). The idea is to detect the landmarks based on their distinct colors. For this purpose an HSV color threshold was defined so a mask can be generated. Once the mask has been generated, a circular blob is used to compute the center of the ball (Myhre et al., 2018; Chavez Armijos et al., 2017). A sample of the color-based feature detection is shown in Figure 7.3.

For the homography-based vision system the same procedure discussed in Section 6.2 is performed together with the innovation filtering explained in Section 6.3.2. A snapshot of the flight test is shown in Figure 7.4.

Additionally, the trajectory followed in this flight test is shown in Figure 7.5. Moreover, the corresponding estimates of the hybrid navigation approach are shown in Figure 7.6, where a comparison is made between the homography-based EKF, the hybrid navigation using an FFI implementation, a dead reckoning solution, and a GPS-based EKF which is

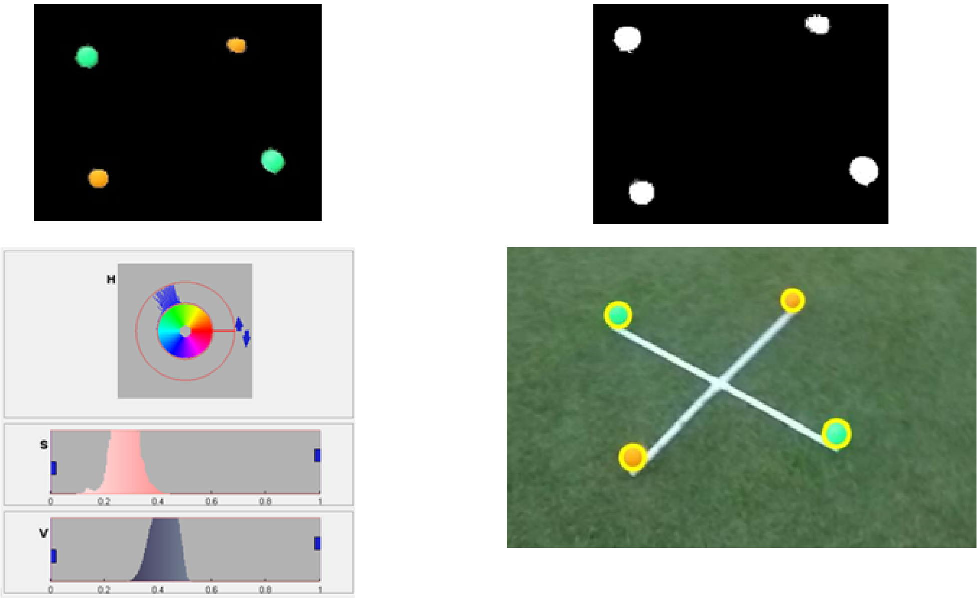


Figure 7.3: Color-Based Landmark Detection Algorithm

Figure 7.4: Hybrid Navigation Test Snapshot

assumed to be the comparison reference. Additionally, a performance RMS error evaluation is presented in Table 7.1. From Figure 7.4 it is possible to see that the error accumulation in the position estimation has been corrected on the hybrid navigation implementation. However, there is an error related to the measurement associated with the known location of the landmark. This error generates a jump in the estimation when the landmark subsystem starts providing a measurement and occurs when measuring the coordinates of the target previous to the flight. However, the overall system tends to stabilize itself after some time. Nevertheless, from Table 7.1, it is possible to observe that the homography-based EKF has a better performance with respect to the hybrid navigation when estimating velocity and attitude. A possible source of error could be poor tuning of the filters, or a propagation of the previously mentioned error during the landmark coordinates measurement.

Table 7.1: RMS Error Hybrid Navigation Comparison Test

		RMS Error		
State	**Units**	**Dead Reckon**	**Homography EKF**	**Hybrid Navigation**
X_N	$[m]$	3319.857	5.979	2.227
Y_N	$[m]$	733.510	4.163	3.719
Z_N	$[m]$	164.037	0.937	1.372
V_{xN}	$[m/s]$	68.335	0.297	0.032
V_{yN}	$[m/s]$	29.064	0.349	0.388
V_{zN}	$[m/s]$	4.743	0.111	0.131
ϕ	$[rad]$	0.0841	0.0193	0.022
θ	$[rad]$	0.0832	0.0147	0.024
ψ	$[rad]$	0.1763	0.0147	0.013

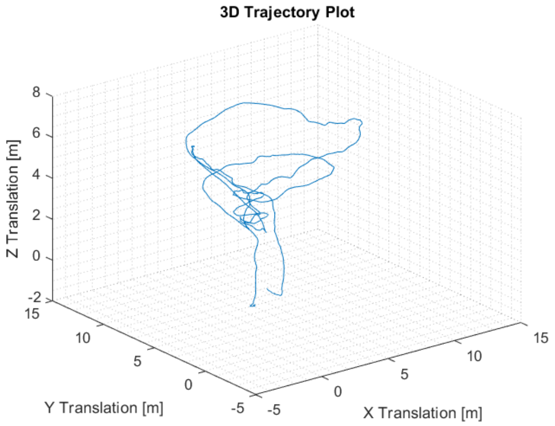
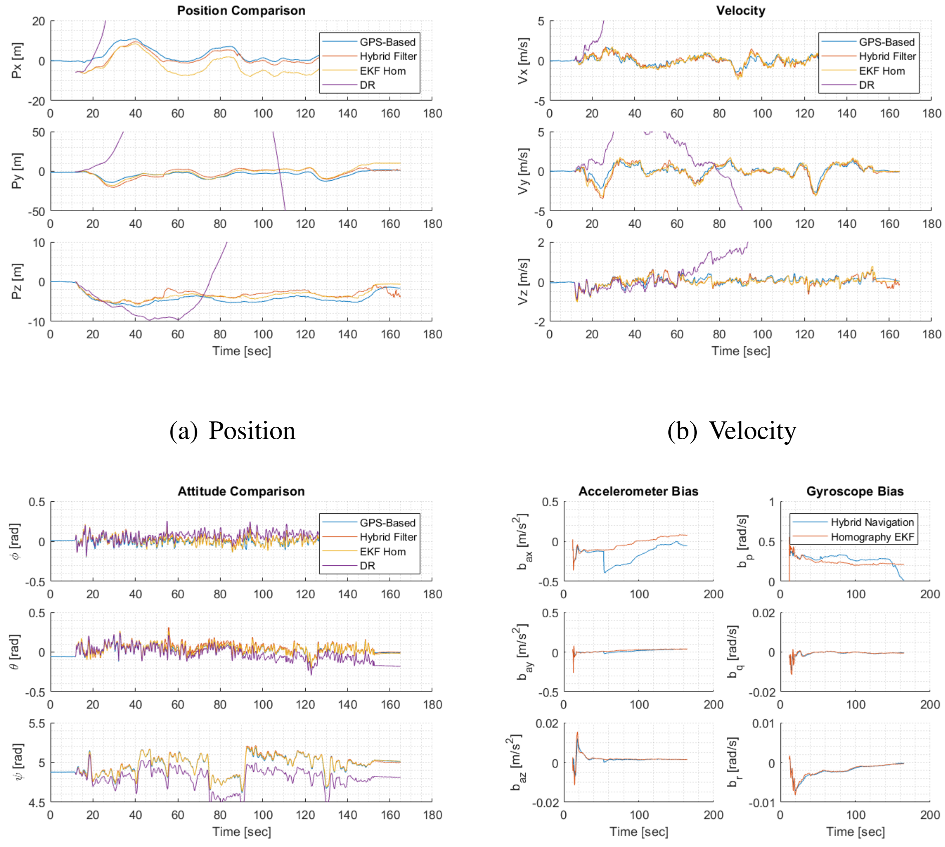


Figure 7.5: Hybrid Navigation Test 3D Trajectory



(a) Position

(b) Velocity

(c) Attitude

(d) Bias

Figure 7.6: Results Hybrid Navigation Flight Test

8. Conclusion

A set of four different vision-based navigation solutions were successfully analyzed via simulation and testing with experimental flight data. These methods were visual odometry, dead reckoning, homography-based Kalman Filter, and a hybrid navigation filter. In addition, two different methods for the disambiguation of the decomposition of the homography matrix were developed and studied. The disambiguation methods enabled the generation of a vision only solution that does depend on an additional altitude sensor to recover the loss of perception typical in monocular systems. However, it was determined that despite the generation of a suitable solution, the visual odometry solution tends to accumulate drift error as a function of time because of the lack of a covariance matrix for error filtering.

A homography-based navigation filter based on inertial measurements (accelerometer and rate gyro data) and the reconstruction of the homography matrix was studied and optimized. The extended Kalman filter includes two different algorithms designed to reject false measurements that typically appear when the camera sensor is close to the ground. In order to test the algorithms a simulation analysis was performed. For the simulation analysis a quadcopter model was developed together with the creation of a virtual environment so the image processing could be tested together with the homography-based Kalman filter. Moreover, a second simulation model was also developed for the generation of a statistical analysis of the vision-based navigation algorithms using a Monte Carlo simulation for comparison. A validation of the results was done by post processing IMU data and camera images obtained from quadcopter flight tests. These flight tests entailed flying an IRIS quadcopter in a circular pattern over a planar region.

In general, the homography-based filter provides an accurate solution for navigation that is independent of GPS measurements. However, the uncertainty in the sigma plots of the simulation shows that the variances of the position in the ”x” and ”y” directions diverge with time, thus generating an error drift in the mentioned states due to the unobservability of these states. This unobservability condition was proven to be true for the stochastic system by performing the rank test on the discrete observability gramian matrix computed for a simulation model. Moreover, by performing an estimability test for observable states, it was determined that the error bias states were less estimable than the other observable states.

During the experimental testing of the filter, some of the assumptions made for the conceptualization of the homography-based EKF were broken to test the limits of the EKF. One test, which entailed flying the quadcopter over a flat but inclined area, showed that the inclusion of a median filter can help with the generation of an approximate solution with better performance than a DR solution. In conclusion, under these tested conditions, the homography-based EKF still generates an improved solution in comparison with the visual odometry and dead reckoning algorithms.

In order to correct the drift from the homography-based EKF, a hybrid navigation system was developed. In the hybrid navigation approach, an additional type of visual measurement was used to make the position states observable. The hybrid navigation uses knowledge of the location of one or more known landmarks in the scene to generate a solution estimate that resets the drift in the homography-based EKF whenever the landmarks are in the field of view of the camera. Additionally, a federated filtered integration (FFI) with a single epoch algorithm was used for the integration of the cascaded measurements as separate independent systems. This FFI implementation resulted in improved accuracy in the position estimation.

9. Recommendations

For the testing of the homography-based navigation, the error covariance that was used assumed that the errors of each of the homography parameters were independent of the other parameters. Therefore, a study of the correlation of the errors between the homography parameters can improve the tuning of the EKF. Additionally, the inclusion of an error model using an AHRS algorithm for the magnetometer can also improve the solutions for heading.

For the observability analysis, the analysis needs to be further extended to the analysis of different conditions due to the coupling of certain states during specific dynamic motions. Additionally, a detectability analysis needs to be performed in order to assess how unobservable are the ”x” and ”y” translational states.

An implementation of the algorithms still needs to be done for further testing of stability. This occurs because of the effects on stability that the processing frequency of the computer can have on the filter itself. This implementation can also serve to perform an analysis of stability on the guidance, navigation, and control of the vehicle by closing the loop.

Since an accurate solution has been generated with the homography-based EKF, a triangulation of the SURF points at two instances of time can be computed so a 3D map of the environment can be generated. A 3D map of the environment can be used for the recognition of new landmarks that would allow the increment of the exploration area when using the hybrid navigation system. The same mapping technique can be used for the homography only EKF solution. However, a consideration of the drift in position needs to be taken into account.

For the hybrid navigation solution, it is necessary to perform a sensitivity analysis that could provide some information about the error correlation effect that the camera has on the two different vision measurements when computing the homography matrix and the landmarks projection with the same image.

References

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359. doi: 10.1016/j.cviu.2007.09.014

Bevly, D. M., & Cobb., S. (2010). *GNSS for vehicle control* (1st ed.). Boston, Mass.: Artech House.

Bradski, G. (2000). The OpenCV Library. *Dr Dobbs Journal of Software Tools*, *25*, 120–125. doi: 10.1111/0023-8333.50.s1.10

Brockers, R., Bouffard, P., Ma, J., Matthies, L., & Tomlin, C. (2011). Autonomous landing and ingress of micro-air-vehicles in urban environments based on monocular vision. *SPIE Defense, Security, and Sensing. International Society for Optics and Photonics*, *8031*, 803111–803111–12. doi: 10.1117/12.884449

Brown, M. (n.d.). *Automatic Panoramic Image Stiching.* Retrieved 2018-06-03, from `http://learning.eng.cam.ac.uk/pub/Public/Turner/Teaching/brown-autostitch.pdf`

Caballero, F., Merino, L., Ferraz, J., & Ollero, A. (2007). Homography based kalman filter for mosaic building. Applications to UAV position estimation. *Proceedings - IEEE*

International Conference on Robotics and Automation(April), 2004–2009. doi: 10.1109/ROBOT.2007.363616

Caballero, F., Merino, L., Ferruz, J., & Ollero, A. (2009). Vision-based odometry and SLAM for medium and high altitude flying UAVs. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *54*(1-3 SPEC. ISS.), 137–161. doi: 10.1007/s10846-008-9257-y

Carlson, N. A. (1990). Federated Square Root Filter for Decentralized Parallel Processes. *IEEE Transactions on Aerospace and Electronic Systems*, *26*(3), 517–525. doi: 10.1109/7.106130

Chavez Armijos, A. S., Myhre, N. J., & Prazenica, R. J. (2017, sep). Vision-Based State Estimation for Asteroid Exploration. In *Aiaa space and astronautics forum and exposition* (pp. 1–17). Winter Park, Florida: American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-5108

Choi, S., Park, J., & Yu, W. (2015). Simplified epipolar geometry for real-time monocular visual odometry on roads. *International Journal of Control, Automation and Systems*, *13*(6), 1454–1464. doi: 10.1007/s12555-014-0157-6

DeAngelo, M. P., & Horn, J. F. (2016). Landmark-Aided Navigation for Air Vehicles Using Learned Object Detectors. In *Aiaa guidance, navigation, and control conference* (pp. 1–17). San Diego, California: AIAA. doi: 10.2514/6.2016-0382 de Visser, M., Schwering, P. B. W., de Groot, J. F., & Hendriks, E. a. (2006). Passive ranging using an infrared search and track sensor. *Optical Engineering*, *45*(2), 026402. doi: 10.1117/1.2173948

Diamant, R., & Jin, Y. (2014). A machine learning approach for dead-reckoning navigation at sea using a single accelerometer. *IEEE Journal of Oceanic Engineering*, *39*(4), 672–684. doi: 10.1109/JOE.2013.2279421

Dogruer, C., Koku, B., & Dolen, M. (2008). Global urban localization of outdoor mobile robots using satellite images. *... Robots and Systems, 2008. ...*, 3927–3932. doi: 10.1109/IROS.2008.4650983

Dubrofsky, E. (2009). *Homography Estimation* (Master of Science, The University of British Columbia). doi: 10.1117/1.3364071

Frese, U., & Hirzinger, G. (2001). Simultaneous Localization and Mapping - A Discussion. *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 17–26.

Garcia Herrera, D. F. (2017). *Design, Development and Implementation of Intelligent Algorithms to Increase Autonomy of Quadrotor Unmanned Missions* (Master Thesis). Embry-Riddle Aeronautical University.

Gong, X., & Zhang, J. (2016). An innovative transfer alignment method based on federated filter for airborne distributed POS. *Measurement: Journal of the International Measurement Confederation*, *86*, 165–181. Retrieved from `http://dx.doi.org/10.1016/j.measurement.2016.02.016` doi: 10.1016/j.measurement.2016.02.016

Groves, P. D. (2008). *Principles of GNSS Inertial and Multi-Sensor Integrated Navigation Systems - GNSS Technology and Applications* (1st ed.). Boston, London: Artech House.

Groves, P. D. (2013). *Principles of Gnss, Inertial, And Multisensor Integrated Navigation Systems* (2nd ed.). Boston, London: Artech House.

Hamza, A. B., Krim, H., & Member, S. (2001). Image Denoising : A Nonlinear Robust Statistical Approach. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, *49*(12), 3045–3054.

Hamza, A. B., Luque-Escamilla, P. L., Martínez-Aroza, J., & Román-Roldán, R. (1999). Removing noise and preserving details with relaxed median filters. *Journal of Mathematical Imaging and Vision*, *11*(2), 161–177. doi: 10.1023/A:1008395514426

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision* (Second ed.). Cambridge Press. doi: 10.1016/S0143-8166(01)00145-2

Hartman, D., Landis, K., Mehrer, M., Moreno, S., & Kim, J. (2014). *Quadcopter dynamic modeling and simulation (quad-sim) v1.00.* Retrieved from `https://github.com/dch33/Quad-Sim`

Hide, C., Botterill, T., & Andreotti, M. (2010). Vision-aided IMU for handheld pedestrian navigation. *Proceedings of the Institute of Navigation GNSS Conference*, 1–9.

Hoffmann, G., Rajnarayan, D., Waslander, S., Dostal, D., Jang, J., & Tomlin, C. (2004). The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC). *The*

23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576), *2*(23), 1–10. doi: 10.1109/DASC.2004.1390847

Hong, S., Chun, H.-h., Kwon, S.-h., Lee, M. H., & Member, S. (2008). Observability Measures and Their Application to GPS / INS. , *57*(1), 97–106. doi: 10.1109/TVT.2007.905610

Indelman, V., Gurfil, P., Rivlin, E., & Rotstein, H. (2010). Navigation Aiding Based on Coupled Online Mosaicking and Camera Scanning. *Journal of Guidance Control and Dynamics*, *33*(6), 1866–1882. doi: 10.2514/1.48134

Indutiveload. (2008). *Radio fix on 1 station.* Retrieved from `https://commons.wikimedia.org/wiki/File:Radio_Fix_on_1_Station.svg` (Accessed: 2018-01-27)

Kaiser, M. K., Gans, N. R., & Dixon, W. E. (2010). Vision-based estimation for guidance, navigation, and control of an aerial vehicle. *IEEE Transactions on Aerospace and Electronic Systems*, *46*(3), 1064–1077. doi: 10.1109/TAES.2010.5545174

Kuemmerle, R., Steder, B., Dornhege, C., Kleiner, A., Grisetti, G., & Burgard, W. (2011). Large Scale Graph-based SLAM Using Aerial Images As Prior Information. In *2008 ieee/rsj international conference on intelligent robots and systems* (Vol. 30, pp. 25–39). IEEE. doi: 10.1007/s10514-010-9204-1

Lee, D., Kim, Y., & Bang, H. (2012). Vision-based terrain referenced navigation for unmanned aerial vehicles using homography relationship. *Journal of Intelligent and*

Robotic Systems: Theory and Applications, *69*(1-4), 489–497. doi: 10.1007/s10846-012-9750-1

Lee, M. H., Park, W. C., Lee, K. S., Hong, S., Park, H. G., Chun, H. H., & Harashima, F. (2012). Observability Analysis Techniques on Inertial Navigation Systems. *Journal of System Design and Dynamics*, *6*(1), 28–44. Retrieved from `http://joi.jlc.jst.go.jp/JST.JSTAGE/jsdd/6.28?from=CrossRef` doi: 10.1299/jsdd.6.28

Lehtinen, A. (2001). *Doppler Positioning with GPS* (Master of Science). Tampere University of Technology.

Lewis, F. L., Xie, L., & Popa, D. (2007). *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory, Second Edition* (Vol. 30) (No. 2). Retrieved from `https://books.google.com/books?hl=en{&}lr={&}id=hNXLBQAAQBAJ{&}pgis=1` doi: 10.1109/MCS.2010.937197

Lin, F., Lum, K.-Y., Chen, B. M., & Lee, T. H. (2009). Development of a vision-based ground target detection and tracking system for a small unmanned helicopter. *Science in China Series F: Information Sciences*, *52*(11), 2201–2215. doi: 10.1007/s11432-009-0187-5

Ma, Y., Soatto, S., Košecká, J., & Sastry, S. (2004). *An Invitation to 3D Vision. 2004* (Vol. 19) (No. 108). doi: 10.1007/978-0-387-21799-6

Mahony, R., Kumar, V., & Corke, P. (2012, sep). Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Robotics & Automation Magazine*, *19*(3), 20–32. doi: 10.1109/MRA.2012.2206474

Mathworks. (n.d.). *Single Camera Calibration App.* Retrieved from https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html

Maus, S., McMillan, S., Mclean, S., Hamilton, B., Thomson, A., Nair, M., & Rollins, C. (2015). The US / UK World Magnetic Model for 2015-2020. *NOAA Technical Report NESDIS/NGDC.*. doi: 10.7289/V5TB14V7

Meyer, J., & Kohbrecher, S. (2014). *Hector quadrotor.* Retrieved from http://wiki.ros.org/hector_quadrotor

Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., & Von Stryk, O. (2012). Comprehensive simulation of quadrotor UAVs using ROS and Gazebo. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7628 LNAI*, 400–411. doi: 10.1007/978-3-642-34327-8_36

Mondragón, I., Campoy, P., Martinez, C., & Olivares-Méndez, M. (2010). 3D pose estimation based on planar object tracking for UAVs control. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 35–41.

Myhre, N., Chavez Armijos, A. S., Clark, M., & Prazenica, R. J. (2018). Vision-Based State Estimation Using Tracked Landmarks. In *2018 aiaa guidance, navigation, and control conference* (pp. 1–17). Winter Park, Florida: AIAA. doi: 10.2514/6.2018-2104

Nütcher, A. (2001). *3D Robotic Mapping The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom* (B. Siciliano, O. Khatib, & F. Groen, Eds.). Springer. doi: 10.1007/978-3-540-73958-6_2

Perez Rocha, A. E., Moncayo, H., Prazenica, R. J., Zacny, K., Mueller, R. P., Dupuis, M., & Ebert, T. (2016). Control Laws Development for a Free-Flying Unmanned Robotic System to Support Interplanetary Bodies Prospecting and Characterization Missions. *AIAA Guidance, Navigation, and Control Conference*(January), 1–23. doi: 10.2514/6.2016-0884

Pink, O. (2008). Visual map matching and localization using a global feature map. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1–7. doi: 10.1109/CVPRW.2008.4563135

Polturi, S. (2007). *Hyperbolic Position Location Estimator with TDOAS from Four Stations* (Doctoral dissertation, New Jersey Institute of Technology). doi: 10.3141/2100-07

Roberts, R., Potthast, C., & Dellaert, F. (2009). Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In *2009 ieee computer society conference on computer vision and pattern recognition workshops, cvpr workshops 2009* (pp. 57–64). Miami, FL, USA: IEEE. doi: 10.1109/CVPRW.2009.5206538

Sabatini R, R. (2013). A Low-cost Vision Based Navigation System for Small Size Unmanned Aerial Vehicle Applications. *Journal of Aeronautics & Aerospace Engineering*, *02*(03). doi: 10.4172/2168-9792.1000110

Stolfi, J. (2009). *Coord system ca.* Retrieved from `https://commons.wikimedia.org/wiki/File:Coord_system_CA_0.svg` (Accessed: 2018-03-05)

Szeliski, R. (2010). *Computer Vision : Algorithms and Applications* (Vol. 5). Springer. doi: 10.1007/978-1-84882-935-0

Titterton, D. H., & Weston, J. L. (2004). *Strapdown Inertial Navigation Technology* (Vol. 17). doi: 10.1049/PBRA017E

Zhang, Z. (2002). A Flexible New Technique for Camera Calibration (Technical Report). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334. doi: 10.1109/34.888718

Zhao, S., Dong, X., Cui, J., Ang, Z. Y., Lin, F., Peng, K., . . . Lee, T. H. (2013). Design and Implementation of Homography-based Vision-aided Inertial Navigation of UAVs. In *Proceedings of the 32nd chinese control conference* (pp. 5101–5106).

Zhao, S., Lin, F., Peng, K., Chen, B., & Lee, T. (2012). Homography-based Vision-aided Inertial Navigation of UAVs in Unknown Environments. *AIAA Guidance, Navigation, and Control Conference*(August), 1–16. doi: 10.2514/6.2012-5033