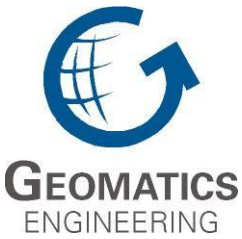
2012-12-14

Vision Based Navigation (VBN) of Unmanned Aerial Vehicles (UAV)

Sheta, Bassem



UCGE Reports
Number 20366

Department of Geomatics Engineering

Vision based Navigation (VBN) of Unmanned Aerial Vehicles (UAV)

(URL: http://www.geomatics.ucalgary.ca/graduatetheses)

by

Bassem Ibrahim Sheta

December, 2012

UNIVERSITY OF CALGARY

Vision Based Navigation (VBN) of Unmanned Aerial Vehicles (UAV)

by

Bassem Ibrahim Sheta

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF GEOMATICS ENGINEERING

and

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

DECEMBER, 2012

Abstract

Autonomous navigation systems used in Unmanned Aerial Vehicle (UAV) are mostly dependent on Global Positioning System (GPS) as a primary means of aiding Inertial Navigation Systems (INS) for accurate and reliable navigation. GPS, however, has limitations in terms of indoor availability and expected signal interference in the GPS-denied environments.

The motivation of this thesis is to address the development of a low cost navigation system used onboard UAVs while maintaining accurate navigation. Motivated by the new advances in visual sensor solutions in combination with traditional navigation sensors, the proposed system is based on fusing visual measurements with INS measurements to achieve comprehensive, fast, real-time, and low cost Vision Based Navigation (VBN) system for the UAV.

VBN is based on localizing set of features (with known coordinates) on the ground and finding their matches in the image taken by an imaging sensor on the UAV using a scale and rotation invariant image matching algorithm. Through Collinearity equations, object space transformation parameters are then estimated such that these matches are transformed into position information. Detailed system design and performance analysis are presented where scenarios include high dynamics of the UAV and different GPS outage are introduced.

To insure fast and robust image matching algorithm, modified Speeded Up Robust Features (SURF) is introduced. The proposed algorithm is implemented on General Purpose (GP) Graphics Processing Unit (GPU) using Compute Unified Device Architecture (CUDA).

Moreover, the developed algorithm is compared against the traditional least square approach, with nonlinear least squares approaches for solving the collinearity equations where large tilted aerial

platform is expected, to overcome the expected nonlinearity of the mathematical model of Collinearity equations.

The navigation solution is then achieved by fusing the vision measurements to the Extended Kalman Filter (EKF) as Coordinate UPdaTe (CUPT) update for the INS measurements.

Performance assessments results demonstrate the enhanced performance of the proposed system against stand-alone INS solutions during a GPS signal outage.

ACKNOWLEDGEMENTS

I wish to express my gratitude to my supervisor, Professor Naser El-Sheimy, who was abundantly helpful and offered invaluable assistance, support and guidance. I was very pleased to attend his great graduate course. His professional teaching is amazing and makes the scientific material interesting and easy for understanding.

Deepest gratitude is also due to my co-supervisor Dr. Mohammed Elhabiby for his vision, proposed ideas, valuable discussion and constructive suggestions during my studies.

Special thanks also to Professor Ayman Habib for his professional teaching and valuable discussion. He helped me to learn and understand photogrammetry which helped me in my research thesis.

I would like also thank Professor Abu Sesay for his helpful suggestions during the initial stages of this work.

Special thanks to my colleagues in my research group, Mobile Mapping Sensor Systems, for their wonderful friendship, support and valuable discussion: Dr. Sameh, Dr. Walid, Dr. Zainab, Dr. Wes, Ahmed El-Ghazouly, Dr. Yigiter, Sara, Adel, Abdelrahman, Mohamed Ali, Mazen, Naif, Siddharth, Dr.Dina and Hsiu-Wen Chang (Killy). It was pleasure to work with them in this wonderful working environment.

I can’t find words to express my gratitude to my MTC colleagues, Dr. Hassan Elhifnawy, Dr. Ahmed Mohsen, Dr. Walid Elthalabawy, Ahmed Shawky and Essam Hamza for their wonderful friend ship and great time we spent together with ourselves and with our families, I really enjoyed each and every moment.

I wish to express my love and gratitude to my beloved parents; although you are thousands of miles away, you were always there whenever I needed you.

This thesis would have never been possible without my loving wife Noha. You were always around at times I was desperate and thought that it would be impossible to continue.

My daughter Malak and my son Yassin, I owe you lots and lots of fun hours. I could not imagine having done my PhD studies without you; you really gave me the reason to go on.

Finally I would like thank my country, Egypt, for funding and supporting my research, specially the Egyptian Armed Forces, my sponsor, for the unwavering support. I also would like to thank my Supervisor Dr. Naser El-Sheimy for the additional funding from his Canada Research Chair and TECTERRA funding.

DEDICATION

To My Beloved Family

My Parents,

My Brothers,

My Lovely Wife, My Daughter and My Son

(Thank You So Much)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES AND ILLUSTRATIONS

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Abbreviation	Definition
BMA	Block Matching Algorithm
BFGS	Broyden, Fletcher, Goldfarb, and Shanno
BPC	Bounded Partial Correlation
CUDA	Compute Unified Device Architecture
CUPT	Coordinate UPdaTe
DCM	Direction Cosine Matrix
DEM	Digital Elevation Model
DoG	Difference of Gaussian
EKF	Extended Kalman Filter
EOP	Exterior Orientation Parameters
FBO	Frame-Buffer Objects
GCP	Ground Control Point
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GP GPU	General Purpose Graphics Processing Unit
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LLF	Local-Level Frame
MAV	Micro-Air Vehicles
NASA	National Aeronautics and Space Administration
NCC	Normalized Correlation Coefficient
NED	North-East-Down
NMS	Non Maximum Suppression
PCA	Principal Component Analysis
PPM	Part Per Million
RF	Radio Frequency
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
UAV	Unmanned Aerial Vehicle
VBN	Vision Based Navigation

LIST OF SYMBOLS

Symbol	Definition
$A_{kinematic}$	Kinematic acceleration
a^b	Accelerometer bias error
a^{drift}	Accelerometer drift error
a^s	Accelerometer scale factor error
β	Set of points defining the trust region
C_l^e	Local to earth frame direction cosine matrix
d_c	Trust region dogleg Cauchy step
d_{GN}	Trust region dogleg Gauss Newton step
d_k	Search direction vector
D_{xx}	Box filter approximation in the x direction
D_V	Interest point descriptor vector
e_i	Circular integrals of pixel intensities
e	First eccentricity of the reference ellipsoid
f	Specific force
$f_{lps}(x_k)$	Linear predicted sum of squares of objective function
$F(x)$	Objective function
G	Gravitational acceleration
g^b	Gyroscope bias error
g^{drift}	Gyroscope drift error
g_{xx}	Second derivative gray level function in x direction
g^s	Gyroscope scale factor error
h	Height
$\mathcal{H}$	Hessian matrix
I	Input image
J	Jacobian matrix
K	Kalman filter gain
L_{xx}	Convolution of the Gaussian second order derivative with input image
M	Meridian radius of curvature
m_k	Trust region model
$n(t)$	Measurements noise
N	Radius of curvature of the reference ellipsoid
P	Position vector
q	Quaternion

Q_k	System noise covariance matrix
$R_{bb}(\tau)$	Autocorrelation function of the signal b
R_k	Measurement noise covariance matrix
R_N	Radius of curvature in the prime vertical
s	Image scale
S	Skew symmetric form of the small incremental changes of rotation over the time
s_k	Trust region step
u	Image measurements vector
ω_e	Earth rotation rate
φ	Geodetic latitude
λ	Geodetic longitude
ϕ	Euler angle roll
θ	Euler angle pitch
ψ	Euler angle heading
Ω_{ib}^{b}	Inertial to body skew-symmetric angular rate matrix
V	Velocity vector
w	Relative weight of the box filter response
W	Process noise
x	Object space transformation parameters to be estimated
$\tilde{\theta}$	Compensated angular increments
$\Delta\tilde{v}$	Velocity increments
ζ	normal gravity
σ	Gaussian scale
(x_p, y_p)	Image coordinates
(X_P, Y_P, Z_P)	object database ground coordinates
(x_{pc}, y_{pc}, f)	Interior orientation parameters
$(X_{PC}, Y_{PC}, Z_{PC}, \omega, \varphi, \kappa)$	Exterior orientation parameters
Δ_k	trust region radius
$\|.\|_k$	iteration dependent norm
$\eta_1, \eta_2, \gamma_1, \gamma_2$	set of constants for trust region algorithm
$\rho_r, \chi, \gamma_c, \xi$	Reflection, expansion, contraction, and shrinkage coefficients for Nelder-Mead algorithm
Φ	Transition matrix

Chapter One: **INTRODUCTION**

In recent years, the utility of the Unmanned Aerial Vehicle (UAV) has greatly increased in applications such as, surveillance, law enforcement and aerial mapping. Furthermore, UAVs are quickly becoming an integral part of both military and commercial operations. For these classes of applications, accurate UAV navigation is considered a critical element for the safe operations of the vehicle. UAV payload is a critical factor which controls the mission capabilities of the UAV. UAV missions include infrastructure and asset management, aerial mapping, post disaster damage assessment, agriculture and environmental monitoring, construction and mining and reconnaissance and weapon delivery.

In recent years there has been an explosion in the number, type and diversity of system designs and application areas of UAVs. However, generally speaking all UAVs typically share the following major component:

(a) Navigation component.

(b) Remote sensing component.

Table 1.1 lists the characteristics and applications of some of the systems currently exist in the market.

Table 1-1: Examples of existing UAVs and their hardware structure

UAV model name	Navigation sensors			Remote Sensing Sensors			Resolution (MP)	Applications
	GNSS	INS	Other	Camera	Laser	other		
Dragonflyer X8	√	√	√	√	----	---	10	Aerial photography ,law

								enforcement, and military tactical surveillance
Gatewing X-100	√	√	√	√	---	---	10	Vegetation monitoring, infrastructure mapping, and topographic surveying
MD4-1000	√	√	√	√	---	---	12	Aerial photography, oil and gas pipeline inspection, fire scene inspection, and law enforcement
LP960	√	√	√	√	---	---	16	Aerial mapping, DEM generation,

								reconnaissance and security
Aerosonde Mark 4.7	√	√	√	√	√	---	N/A	Intelligence, surveillance and reconnaissance.

To date, most UAV navigation systems rely mainly on the Global Positioning System (GPS) receivers as the primary source of information to provide the position of the vehicle. GPS is able to provide precise positioning information to an unlimited number of users anywhere on the planet. GPS, however, can provide these types of information only under ideal conditions which require an open environment (i.e. open space areas). In other words, the system doesn't work very well in urban, canopy areas due to signal blockage and can be totally blocked if the signal is jammed. More recently, and accepting that these techniques must inevitably cost more than a GPS-based system alone, the concept of combining complimentary navigation systems has been accepted for UAV military and some commercial applications.

1.1 Problem statement

Positioning, location, and navigation are distinct processes in these systems. Positioning is determining coordinates in a fixed coordinate system (Krakiwsky and Wells 1971). Coordinates by themselves do little to tell the average person where they are. Location is the process of putting coordinates into a frame of reference that is more useful. Typically, a map with topographical features, navigational aids, roads, street addresses, etc. is used to determine location (Krakiwsky 1995). Navigation regards the provision of directional information on how

to travel from the vehicle's present position to a given destination or a set of waypoints (Harris 1989).

The integration of INS with its Inertial Measurement Unit (IMU) and GPS provides a system that has superior performance than each individual system. For instance, GPS derived positions have approximately white noise characteristics over the whole frequency range. The GPS-derived positions and velocities are therefore excellent external measurements for updating the INS with position parameters, thus improving its long-term accuracy. Similarly, the INS can provide precise position and velocity data for GPS signal acquisition and reacquisition after outages. In general, the fact that redundant measurements are available for the determination of the vehicle trajectory parameters greatly enhances the reliability of the system (Nilsson 2005) specially during GPS signal jamming which typically happens in military environments. Although the INS/GPS configuration is considered as a complete navigation system, it has always a main drawback when GPS signal are blocked or jammed which occur during navigation. In this case, navigation is provided by the INS only until GPS signals are obtained again. Due to the known time-dependent error behavior of the INS, the obtained positions errors during such periods drift rapidly with time. This will be dominant in the case of low-cost inertial sensors. Therefore, although the INS can provide positions during GPS outages, the accuracy of such positions cannot meet the requirements of several navigation applications. Hence, methods for improving the navigation solution should be applied.

Taking advantage of the already installed onboard visual sensors for the purpose of safely navigating the UAV, through Vision Based Navigation (VBN) to complete the mission is one of the options.

Emerging visual sensor solutions show a lot of promise in replacing or improving the traditional IMU or GPS system in many mission scenarios (Stepanyan 2006). The basic concept of VBN is to localize a set of features (with known coordinates) on the ground and find their matches in the image taken by the imaging sensor on the UAV. Then, through the use of photogrammetric techniques (e.g. the collinearity equations), this match is transformed into position information. Finally, this information is integrated with the INS measurements and work as a CUPT for the aerial vehicle. This thesis aims to develop and implement a comprehensive fast and real-time low cost VBN methodology for UAVs. One of the critical components in VBN is the robustness of the matching algorithm between the images captured by the UAV and the set features on the ground. A robust scale and rotation invariant image matching algorithm is important for VBN of aerial vehicles. This will be done by the modified SURF algorithm, where matches between the geo-referenced database images and the new real-time captured ones are determined.

Another important aspect of VBN is the proper estimation of the object space transformation parameters which improves the efficiency of the navigation process by the real-time estimation of transformation parameters used in aiding the inertial measurements data. Different methods are introduced to solve the collinearity equations which describes the 2D-3D transformation model. These methods are Trust region, Trust region dogleg algorithm and Levenberg-Marquardt, Nelder-Mead Simplex Direct Search, Quasi-Newton Line Search.

Finally, all the above information has to be integrated into an estimation filter which can accommodate the navigation sensors data and the position/coordinate update (CUPT) information from the image matching to provide the optimal navigation information for the UAV. This will be accomplished through fusing the vision measurements to the EKF as CUPT update for the INS measurements employing the residuals originating from the object space

transformation parameters estimation as the error covariance matrix in the EKF implementation. Consequently, the UAV position, velocity, and attitude are estimated during GPS outage from vision measurements when available.

1.2 Background

(Hagen and Heyerdahl 1992) proposed an approach for estimating an aircraft's position and orientation using visual measurements for landmarks located on a known topographic map using an EKF. In this approach, landmarks, referred to as "tokens", are detected based on maximizing a uniqueness measure that prevents such tokens from being too close to each other as the terrain around them is linear. The uniqueness measure detects the point of interest in the matching algorithm based on the spatial distance and feature distance between points of interest candidates. Then, those tokens are described based on circular integrals of pixel intensities:

$$e_i(P) = \int_0^{2\pi} f(x + r_i \cos\alpha, y + r_i \sin\alpha) d\alpha \qquad \textbf{1.1}$$

Where *P(x,y)* is a point in the image (Hagen and Heyerdahl 1992). Such descriptors are invariant for translation and rotation.

Another approach for estimating aircraft position and velocity from sequential aerial images was proposed by (Sim et al. 1996; Dong-Gyu et al. 2002). The method presented by (Dong-Gyu et al. 2002) provides real-time implementation of a vision based navigation algorithm which accomplishes both accuracy and effectiveness (in other words, the low quality of the sensors used, computational load and complexity). The new algorithm is composed of two sections: relative and absolute position estimation, which are connected to each other through a switching scheme.

The relative position estimation section is essentially based on the stereo modeling of two sequential images where the feature point of a current and previous image is utilized for extracting the displacement of the aircraft. This is achieved by applying the Block Matching Algorithm (BMA) and Normalized Correlation Coefficient (NCC) where two levels of Gaussian based hierarchical matching are used to lower the computational load of the algorithm. Accumulation of the displacement calculation leads to a position measurement for the aircraft. The velocity of the aircraft is then obtained by dividing these displacements by the sampling interval time. However, accumulating these displacement measurements yields errors in navigation parameters estimation subsequently increasing with time as well.

The next step involves estimating the absolute position which corrects the errors arising due to the accumulation of displacement measurements performed through relative position estimation. This is achieved through matching schemes using reference images (if the effective range from the reference position is 400 m and distinct artificial landmarks are available in the scene) and Digital Elevation Model (DEM) (if the effective range is 200m and areas where no artificial landmarks are available).

Another approach for estimating aircraft position from image sequences was proposed by (Lerner et al. 2004). In this approach, aircraft position and orientation was estimated from combining optical flow measurements, acquired from two successive images, with DTM. The optical flow is defined as an array vector which includes the correspondence vectors between two successive images. The approach starts with calculating the optical flow vector and then compares it to the scene DTM. Velocity measurements from the optical flow can be integrated to provide position estimates. However, this is subject to error drift over time. To eliminate this drift, the optical flow measurements are integrated with the scene DTM to estimate the aircraft

position relative to the earth. The optical flow approach is considered a grid-based approach which divide the image into a grid of image patches while ignoring image intensity quality (Barron et al. 1992; McCarthy and Barnes 2004). This leads to difficulty in locating correspondence interest areas between image frames.

(Olson et al. 2000) proposed incorporating stereo imaging sensor for navigation and obstacle avoidance. This approach starts with selecting initial landmarks necessary to start up the 3D position estimate. The initial landmarks are selected from the left image of the first pair of successive images. Stereo matching using correlation is then applied to the first pair. The previously calculated relative position between the two cameras is used to estimate the position of the landmarks with respect to the camera frame. However, the correlation based matching technique is scale and rotation variant which make it not suitable for navigation.

Object detection and avoidance for aerial vehicles was addressed in (Bhanu et al. 1990; Roberts and Bhanu 1992). The proposed approach fused inertial measurements with information that originated from image sequences to calculate range measurements for estimating the object distance.

The algorithm consists of the following steps based on two frames taken at times t_1 and t_2:

1. For the two frames, calculate the navigation state of each image using the inertial data.
2. Extract interest points from each frame.
3. Locate the focus of expansion using the velocity vector from inertial measurements.
4. Project the focus of expansion and interest points in the second frame onto an image plane parallel to the first frame.
5. Match the interest points from the second frame to the interest points from the first frame.
6. Compute the range to each interest point.

7. Create dense range maps using the computed range values to obstacles.

Interest points are detected using the Hessian and Laplacian operators according to the following:

$$I(g) = g_{xy}^2 - g_{xx}g_{yy} \qquad \textbf{1.2}$$

Where g_{xx} is the gray level function and is the second derivative in the x direction (Roberts and Bhanu 1992).

This approach however, was just an initial solution for integrating inertial with vision measurements to help obstacle avoidance. It showed the importance of using inertial measurements to help solve the correspondence problem (Veth 2006).

Another application that uses the augmentation of inertial measurements with image-based motion estimation was presented in (Roumeliotis et al. 2002). This approach was proposed for helping NASA missions achieve accurate and safe landing on planetary bodies.

The sensors used in this algorithm are INS, laser altimeter, and image sensor. Additionally, the applied image-based motion estimation approach can be categorized as a two-frame feature based motion estimation. The measurements originating from those sensors are fused through a modified Kalman filter, which estimates the errors in the estimated states, for vehicle navigation.

(Pachter and Porter 2004) proposed an approach for tracking unknown ground objects using INS aiding. Using the driftmeter concept, a mounted telescope with a gimbal onboard the aircraft is used to measure angles and angular rates with respect to fixed ground landmarks. The proposed algorithm has assumptions which include: constant angle of attack, constant speed, and straight flight path. Given these assumptions, initial deflection angle and angular rate from the driftmeter

are used to estimate the aircraft velocity. The aircraft angle of attack is then measured using the angle from the aircraft velocity vector to the initial Line Of Sight (LOS) to the target. The navigation solution is then estimated using the angle of attack measurements and INS measurements using weighted least squares.

(Raquet 2003) proposed a tightly coupled, feed-forward Kalman filter approach to estimate the navigation solution for aircraft combining visual measurements of landmarks with inertial measurements and barometric altimeter measurements. The proposed algorithm assumes that landmarks registration algorithm, which identifies landmarks appearance in successive images, uses aircraft attitude and positions coming from the INS measurements. Such assumption cannot be used in the case of low cost INS.

1.3 Research objectives and contributions

1.3.1 Objectives

The major objective of this thesis is to develop and implement real-time VBN technique for UAV. The implementation will be tested through simulated visual measurements combined with simulated GPS/INS paths at low altitude.

The thesis will include several sub-objectives that will serve the major objective. The sub-objectives will focus on the development of three main components that will be integrated to form the final VBN technique. The three components are: (a) the development of fast image matching algorithm based on SURF, (b) the development of object space transformation parameters optimization algorithm for estimating the position of the vision sensor, and (c) the development of coordinate fusion through EKF for the optimal estimation of the UAV navigation state. To achieve the overall objective of this thesis, the following sub-objectives will be addressed:

1. Implementing a fast SURF-36 algorithm for image matching between the database images and UAV real-time captured images.
2. Developing a fast SURF algorithm with short descriptor of 3×3 sub regions (SURF-36) for fast image matching of the VBN.
3. Investigating and implementing a different samples count in the sub-divisions of the different types of SURF algorithm (SURF 36, SURF 64, and SURF 128) and to test the effect of the number of samples in each subdivision on the accuracy of the matching algorithm.
4. Implementing the developed algorithms on CUDA platform. The use of CUDA satisfies the real-time processing requirements, while keeping the high accuracy which is suitable for VBN real-time implementation.
5. Assessing different nonlinear optimization algorithms for estimating the object space transformation parameters to overcome the nonlinearity originating from the expected large transformation parameter.
6. Developing an efficient EKF fusion approach to integrate the position estimated from the image matching with inertial measurement for the optimal estimation of the UAV's navigation state.
7. Developing a simulation software package for testing and assessing of the developed algorithms.

1.4 Thesis outline

The work in this thesis is structured as the following:

- Chapter two focuses on the concepts and background necessary for introducing the INS/Vision measurement fusion problem. The chapter starts with the navigation

reference frames and the necessary transformation between these different frames. Next, based on the navigation frame, the inertial navigation equations are introduced. Finally, the INS mechanization equations are derived.

- Chapter three reviews the concepts and background of image matching and its key role in the VBN. The chapter will also introduce the modified SURF algorithm as the image matching algorithm for VBN. The chapter starts with a literature review of the most important research work done in image matching techniques. Next, the SURF algorithm will be explained starting from the interest point detection to SURF descriptors building and finally implementation of the modified SURF algorithm using Graphics Processing Unit (GPU) Compute Unified Device Architecture (CUDA) to meet the real-time processing requirements of UAVs.
- Chapter four introduces and investigates the estimation of the georeferencing parameters necessary for VBN of aerial vehicles for large values of the rotational angles, which will lead to nonlinearity of the estimation model. In this case, traditional least squares approaches will fail to estimate the georeferencing parameters, because of the expected nonlinearity of the mathematical model. Five different nonlinear least squares methods are presented which include four gradient based nonlinear least squares methods (Trust region, Trust region dogleg algorithm, Levenberg-Marquardt, and Quasi-Newton line search method) and one non-gradient method (Nelder-Mead simplex direct search).
- Chapter five introduces an alternative method for UAV navigation based on visual measurements in GPS-denied environments. The system applies the concept of photogrammetric resection to update the IMU measurements, which is achieved by finding correspondences between real-time images captured during flight mission and

geo-referenced images in the database. Then, the collinearity equations are used to estimate the position of the perspective center of the camera through the concept of photogrammetric resection. The object space transformation parameters, the output of the photogrammetric resection, are used as the update for the INS kalman.

- Chapter six concludes the results of this thesis and provide recommendations for future research and further improvements of the developed methodology in the thesis.

Chapter Two: **NAVIGATION WITH INERTIAL NAVIGATION SYSTEMS**

This chapter will provide the INS mechanization equations in the Local Level frame (l-frame) which can be applied to low grade IMUs. Different coordinate frames will be defined and the transformation between them is introduced in the first section. In the second section, an introduction of inertial positioning is introduced. Next, based on the navigation frame, the navigation equations are introduced. Finally, the INS mechanization equations are introduced.

2.1 Coordinate reference frames and transformations

Reference frames, their angular rates with respect to one another, and the transformations between them, will be frequently required throughout the navigation equations of inertial systems. INS development and analysis include many reference frames to properly express aircraft dynamic characteristics such as position, velocity, and orientation. The four coordinates frames most frequently used are:

1. The operational inertial frame (i-frame).
2. The conventional terrestrial frame (e-frame).
3. The local-level frame (ℓ-frame).
4. The body frame (b-frame).

2.1.1 Operational inertial reference frame (i-frame)

The Operational inertial frame, denoted by 'i', is non-rotating and non-accelerating with respect to a true inertial frame within the accuracy of the measurements used to define it. For the applications considered here, the definition of the operational inertial frame is as follows:

1. origin : at the center of mass of the Earth,
2. Z-axis: parallel to the instantaneous spin axis of the Earth.

3. X-axis: pointing towards the mean equinoctial colure,

4. Y-axis: completing a right-handed orthogonal frame.

Because of the definition of the origin, this system is strictly speaking an approximation to a quasi-inertial system. The physical realization of such a frame is given by a star catalogue system, a system of quasar sources, or, to a lower accuracy, by a system of gyroscopic axes.

2.1.2 Conventional terrestrial frame (e for Earth-fixed frame)

The Conventional Terrestrial Frame, denoted by 'e' in Figure 2.1, is an Earth-fixed Cartesian frame, defined as follows:

1. Origin: Earth's center of mass.
2. Z-axis: parallel to the mean spin axis of the Earth.
3. X-axis: pointing towards the mean meridian of Greenwich.
4. Y-axis: orthogonal to the X and Y axes completing a right handed orthogonal frame.

It will be assumed in the following that within the approximations given here, the i-frame and the e-frame differ only by a constant angular rate, equal to the mean rotation of the Earth i.e., about a common z-axis.

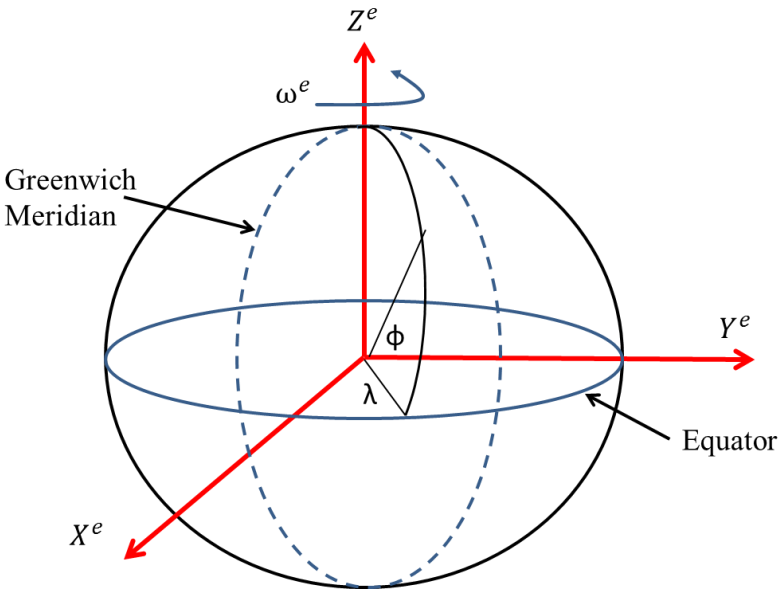


Figure 2.1: Definition of the e-frame

The vector of angular velocities describing the rotation of the e-frame with respect to the i-frame projected to the e-frame is written as follows:

$$\omega_{ie}^{e} = \begin{bmatrix} 0 \\ 0 \\ \omega_e \end{bmatrix} \qquad \textbf{2.1}$$

where:

ω_e is the earth rotation rate.

To describe a position vector in the e-frame P^e with respect with the geodetic latitude (φ), longitude (λ), and height (h), the following formula is used (Jekeli 2001) :

$$P^e = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (R_N + h)\cos\varphi\cos\lambda \\ (R_N + h)\cos\varphi\sin\lambda \\ (R_N(1-e^2) + h)\sin\varphi \end{bmatrix} \quad \textbf{2.2}$$

where:

e is the first eccentricity of the reference ellipsoid.

R_N is the radius of curvature in the prime vertical.

2.1.3 Local level reference frame

The local level frame (l-frame) is a local geodetic frame and is defined as follows:

1. Origin: at a predefined point on the vehicle or the sensor onboard the vehicle.
2. X-axis: is in the direction of the geodetic north.
3. Z-axis: perpendicular to the reference ellipsoid pointing down.
4. Y-axis: completes the right handed orthogonal frame, i.e. the north-east-down (NED) system as shown in Figure 2.2.

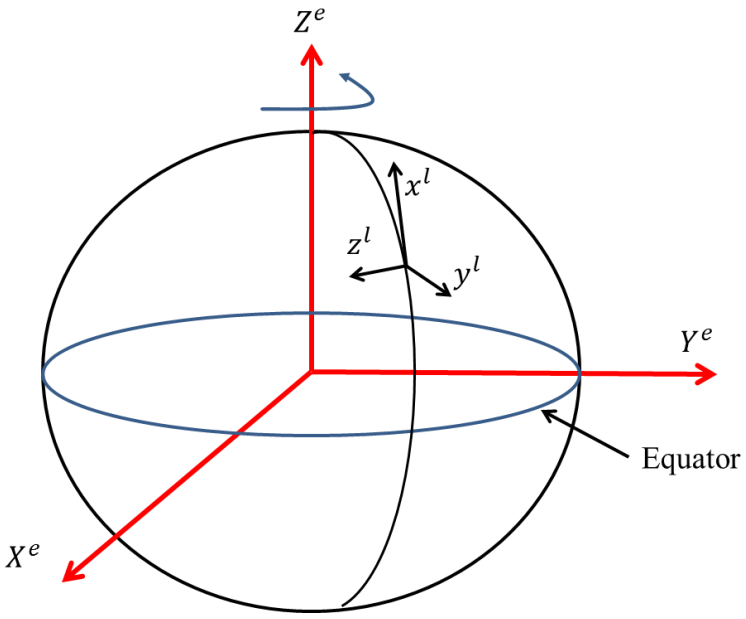


Figure 2.2: the e-frame and l-frame representation (Jekeli 2001)

The Direction Cosine Matrix (DCM) is used to describe the transformation from the l-frame to the e-frame in terms of the geodetic latitude (φ) and longitude (λ) according to the following:

$$C_l^e = \begin{bmatrix} -\sin\varphi\cos\lambda & -\sin\lambda & -\cos\varphi\cos\lambda \\ -\sin\varphi\sin\lambda & \cos\lambda & -\cos\varphi\sin\lambda \\ \cos\varphi & 0 & -\sin\varphi \end{bmatrix} \qquad \textbf{2.3}$$

The quaternion corresponding to C_l^e is described as:

$$q_l^e = \begin{bmatrix} \cos(-\pi/4 - \varphi/2)\cos(\lambda/2) \\ -\sin(-\pi/4 - \varphi/2)\sin(\lambda/2) \\ \sin(-\pi/4 - \varphi/2)\cos(\lambda/2) \\ \cos(-\pi/4 - \varphi/2)\sin(\lambda/2) \end{bmatrix} \qquad \textbf{2.4}$$

To describe the Earth rotation rate vector in the l-frame, the following equation is used:

$$\omega_{ie}^{l} = C_{e}^{l}\omega_{ie}^{e} = \begin{bmatrix} \omega_e \cos\varphi \\ 0 \\ -\omega_e \sin\varphi \end{bmatrix} \quad \textbf{2.5}$$

(Titterton and Weston 1997) described the rotation rate vector of the l-frame with respect to the e-frame in terms of the rate of change of the latitude and longitude as:

$$\omega_{el}^{l} = \begin{bmatrix} \dot{\lambda}\cos\varphi & -\dot{\varphi} & -\dot{\lambda}\sin\varphi \end{bmatrix}^{T} \quad \textbf{2.6}$$

2.1.4 The body reference frame

The body frame (b-frame) is an orthogonal frame whose axes coincide with the axes of the IMU mounted on the aircraft. It is installed in such a way that it represents the forward (*x*), transversal (*y*), and downward (*z*) axes of an aircraft. It is defined as follows:

1. Origin: at the center of the IMU.
2. The axes are aligned with the nose, right wing, and bottom of the UAV as shown in Figure 2.3.

Figure 2.3: UAV body frame.

The transformation from the l-frame to the b-frame can be expressed as (Titterton and Weston 1997):

$$C_l^b = R_x(\phi)R_y(\theta)R_z(\psi) \tag{2.7}$$

where:

ϕ is the Euler angle roll.

θ is the Euler angle pitch.

ψ is the Euler angle heading.

Therefore, using orthognality, the DCM from the b-frame to the l-frame can be calculated as:

$$\begin{aligned} C_b^l &= (C_b^l)^T = R_z(-\psi)R_y(-\theta)R_x(-\phi) \\ &= \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \\ &= \begin{pmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{pmatrix} \end{aligned} \tag{2.8}$$

The Euler angels can then be extracted from the DCM C_b^l as follows:

$$\theta = -\tan^{-1}\left(\frac{c_{31}}{\sqrt{1-c_{31}^{\ 2}}}\right) \tag{2.9}$$

$$\phi = a\tan 2(c_{32}, c_{33}) \tag{2.10}$$

$$\psi = a\tan 2(c_{21}, c_{11}) \tag{2.11}$$

where:

c_{ij}, $1 \le i, j \le 3$, are the DCM C_b^l elements.

2.2 Inertial navigation

Inertial navigation is defined as the computation of position and velocity using initial position and velocity and the time change of the vehicle acceleration. The first integral of the vehicle acceleration plus the initial velocity are used to compute the vehicle velocity while the vehicle position is computed using the initial position plus the integral of velocity over time. The integration processes are computed in the Earth-fixed frame while measurements frame is defined by the IMU axes. In the following, the strapdown concept will be treated, where the Earth-fixed frame (chosen as l-frame) is established numerically inside the IMU.

INS measurements are computed in the i-frame. Consequently, rotational dynamics between the b-frame of the IMU, the i-frame, and the Earth-fixed frame are necessary to derive the vehicle acceleration.

Three accelerometers and three gyroscopes mounted on an orthogonal triad are used to determine the aircraft navigation states (position, velocity, and attitude). The accelerometers measure the specific force as:

$$\mathbf{f} = A_{kinematic} - G \qquad \textbf{2.12}$$

Where:

$\mathbf{f}$ is the specific force.

$A_{kinematic}$ is the kinematic acceleration.

G is the gravitational acceleration.

The specific force measurements are expressed in the i-frame which is different from the frame which velocity and position are usually expressed (l-frame). Therefore, the IMU includes the gyro triad, which provide angular change with respect to the initial orientation, to transform the

measured specific force in the l-frame.Therefore, inertial positioning is achieved by combining rotation and specific force measurements, which provide estimates of velocity, position, and attitude of the moving aircraft with respect to the l-frame.

Newton's laws can be used to design inertial sensors, which are devices that measure linear acceleration and angular velocity with respect to an inertial frame of reference. Many of these sensors are rebalancing devices measuring either the force or the torque needed to maintain a given linear or angular momentum. They can therefore be viewed as realizations of the fundamental conservation principles of physics.

2.3 Inertial navigation equations

Considering a position vector P^i, the gravity acceleration G^i, the specific force $\mathbf{f}^i$, and Newton's laws, we can deduce an equation for the kinematical acceleration:

$$\ddot{P}^i = G^i + \mathbf{f}^i \qquad \textbf{2.13}$$

If two coordinate reference frames have rotational angular velocity ω relative to each other, rotation matrix between these two frames is composed of a set of time variable equations. A set of differential equations is used to describe the time change of the rotation matrix. Assuming a position vector of a point fixed in the b-frame P^b, the transformation of its coordinates from the body frame to the inertial frame is given by:

$$P^i(t) = R_b^i(t)P^b \qquad \textbf{2.14}$$

Considering P^b as constant and differentiating both sides of the above equation with respect to time will lead to:

$$\dot{P}^{i} = \dot{R}_b(t)\, P^b \qquad \textbf{2.15}$$

Using the method of rigid body rotation about a fixed axis, the time derivative of the position vector P can be computed as:

$$\frac{dP}{dt}|i = \frac{dP^b}{dt}|b + \omega_{ib}^i \times P^b = \omega_{ib}^i \times P^b \qquad \textbf{2.16}$$

Since $\frac{dP}{dt}|b = 0$ and $\omega_{ib}^i \times P^b = \Omega_{ib}^i \bullet P^b$, therefore equation 2.15 can be written as:

$$\dot{P}(t) = R_b^i(t) \bullet \Omega_{ib}^b \bullet P^b \qquad \textbf{2.17}$$

From equations 2.15 and 2.17, the differential equations for the rotation matrix are given as:

$$\dot{R}_b^i = R_b^i \Omega_{ib}^b \qquad \textbf{2.18}$$

Equation 2.18 expresses that the time derivative of the rotation matrix is related to the angular velocity vector ω of the relative rotation between the b-frame and i-frame.

Assuming the position vector P^b is time dependent, the time derivative of the position vector is computed by differentiating both sides of equation 2.14:

$$\begin{aligned} \dot{P}^{i} &= \dot{R}_b^i\, P^b + R_b^i\, \dot{P}^{b} \\ &= R_b^i \Omega_{ib}^b P^b + R_b^i\, \dot{P}^{b} \end{aligned} \qquad \textbf{2.19}$$

The terms of the above equation can be arranged as follows:

$$\dot{P}^{i} = R_b^i(\dot{P}^{b} + \Omega_{ib}^b P^b) \qquad \textbf{2.20}$$

This equation describes the transformation of the velocity vector from the b-frame to the i-frame and called the Coriolis equation (after the mathematician Gaspard-Gustave Coriolis).

The time derivative of the velocity vector in equation 2.20, is given as follows:

$$\ddot{P}^{i} = R_b^i(\ddot{P}^{b} + 2\Omega_{ib}^b \dot{P}^{b} + \dot{\Omega}_{ib}^{b} P^b + \Omega_{ib}^b \Omega_{ib}^b P^b) \qquad \textbf{2.21}$$

where:

$\ddot{P}^{i}$ is the body acceleration in the body frame monitored by accelerometers.

Ω_{ib}^b is the body Angular velocities monitored by Gyroscopes.

Solving the vector differential equations will result in a time variable state vector with kinematic sub-vectors for position, velocity, and attitude. Depending on the computational frame used, the results can be represented in the i-frame, the e-frame, or the l-frame, respectively.

The aircraft motion modelling is preferred in the l-frame for the following reasons:

1. L-frame axes are aligned to the local east, north, and up directions leading to computation of the attitude angles (pitch, roll, and azimuth) directly as the output of the mechanization equations.
2. L-frame definition is based on the normal to the reference ellipsoid which leads to computation of the geodetic coordinate differences directly as the output of the system.
3. The computational errors in the navigation parameters on the North-East plane are bounded due to the Schuler effect.

To describe the aircraft motion in the l-frame, the coordinate states are described as the curvilinear coordinates as follows:

$$P^l = (\varphi, \lambda, h)^T \qquad \textbf{2.22}$$

Where:

$\varphi, \lambda, and\ h$ are the latitude, longitude, and height respectively.

The earth-referenced velocity of the aircraft is defined in the l-frame as follows:

$$V^l = R_e^l \dot{P}^e = (V_e, V_n, V_u)^T \qquad \textbf{2.23}$$

The time derivative of the position vector states P^l is expressed with respect to the earth-referenced velocity V^l by differentiating equation 2.22 as follows:

$$\dot{P}^l = D^{-1} V^l \qquad \textbf{2.24}$$

Where:

$$D^{-1} = \begin{pmatrix} 0 & 1/(M+h) & 0 \\ 1/(N+h)\cos\varphi & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ where M and N are the meridian and prime vertical radii of curvature.

To describe the velocity vector in the l-frame, the accelerometers measurements in the l-frame are used. However, the accelerometers measurements in the l-frame cannot provide the velocity vector in the l-frame for the following reasons:

1. The first reason is due to the Earth rotation rate ($\omega^e = 15\,\text{deg.}/hr$). This Earth rotation rate will be expressed in the l-frame as an angular velocity vector $\omega_{ie}^l = (0, \omega^e \cos\varphi, \omega^e \sin\varphi)^T$.

2. The second reason is the change of thel-frame orientation with respect to the Earth. This effect is expressed by the angular velocity vector as:

$$\omega_{el}^{l} = \left(-\dot{\varphi}, \dot{\lambda}\cos\varphi, \dot{\lambda}\sin\varphi \right)^{T} = \left(-\frac{V^{n}}{R+h}, \frac{V^{e}}{R+h}, \frac{V^{e}\tan\varphi}{R+h} \right)$$

3. The third reason is the Earth's gravity field given as $G^{l} = (0, 0, -G)^{T}$.

Consequently, the rate of change of the aircraft velocity can be expressed in the l-frame as:

$$\dot{V}^{\ell} = R_{b}^{\ell} f^{b} - (2\Omega_{ie}^{\ell} + \Omega_{e\ell}^{\ell})V^{\ell} + G^{\ell} \qquad \textbf{2.25}$$

Where Ω_{ie}^{ℓ} and $\Omega_{e\ell}^{\ell}$ are the skew-symmetric matrices corresponding to ω_{ie}^{ℓ} and $\omega_{e\ell}^{\ell}$ respectively and they are expressed as follows:

$$\Omega_{ie}^{\ell} = \begin{pmatrix} 0 & -\omega^{e}\sin\varphi & \omega^{e}\cos\varphi \\ \omega^{e}\sin\varphi & 0 & 0 \\ -\omega^{e}\cos\varphi & 0 & 0 \end{pmatrix}$$

$$\Omega_{e\ell}^{\ell} = \begin{pmatrix} 0 & \frac{-V^{e}\tan\varphi}{N+h} & \frac{V^{e}}{N+h} \\ \frac{V^{e}\tan\varphi}{N+h} & 0 & \frac{V^{n}}{M+h} \\ \frac{-V^{e}}{N+h} & \frac{-V^{n}}{M+h} & 0 \end{pmatrix} \qquad \textbf{2.26}$$

Where: R_{b}^{ℓ} is a direction cosine matrix used to transform the measured specific force vector into navigation axes. This matrix propagates in accordance with the following equation:

$$\dot{R}_{b}^{\ell} = R_{b}^{\ell}\Omega_{\ell b}^{b} \qquad \textbf{2.27}$$

where:

$\Omega^b_{\ell b}$ is the skew symmetric form of $\omega^b_{\ell b}$, which is the body rate with respect to the l-frame. This is derived by differencing the measured body rates ω^b_{ib} and estimates of the components of l-frame rate ω_{il} . The latter term is obtained by summing the Earth's rate with respect to the inertial frame and the turn rate of the navigation frame with respect to the Earth. Therefore,

$$\begin{aligned} \Omega^b_{\ell b} &= \Omega^b_{\ell i} + \Omega^b_{ib} = -\Omega^b_{i\ell} + \Omega^b_{ib} \\ \Omega^b_{\ell b} &= \Omega^b_{ib} - \Omega^b_{i\ell} \end{aligned} \qquad \textbf{2.28}$$

It is instructive to consider the physical significance of the various terms in the navigation equation. From this equation, it can be seen that the rate of change of the velocity, with respect to the surface of the Earth, consists of the following terms:

1. The specific force acting on the vehicle, as measured by a triad of accelerometers mounted within it.
2. A correction for the acceleration caused by the vehicle's velocity over the surface of a rotating Earth, usually referred to as the Coriolis acceleration.
3. A correction for the centripetal acceleration of the vehicle, resulting from its motion over the Earth's surface. For instance, a vehicle moving due east over the surface of the Earth, will trace out a circular path with respect to inertial axes. To follow this path, the vehicle is subject to a force acting towards the centre of the Earth of magnitude equal to the product of its mass, its linear velocity and its turn rate with respect to the Earth.
4. Compensation for the apparent gravitational force acting on the vehicle. This includes the gravitational force caused by the mass attraction of the Earth, and the centripetal acceleration of the vehicle resulting from the rotation of the Earth. The latter term arises

even if the vehicle is stationary with respect to the Earth, since the path which its follows in space is circular (Titterton et al. 2004).

The previous navigation equations can be summarized in one equation describes the position, velocity, and attitude equations:

$$\begin{pmatrix} \dot{P}^{\ell} \\ \dot{V}^{\ell} \\ \dot{R}_b^{\ell} \end{pmatrix} = \begin{pmatrix} D^{-1}V^{\ell} \\ R_b^{\ell}\ f^b - (2\Omega_{ie}^{\ell} + \Omega_{e\ell}^{\ell})V^{\ell} + G^{\ell} \\ R_b^{\ell}(\ \Omega_{ib}^b - \Omega_{i\ell}^b) \end{pmatrix} \qquad \textbf{2.29}$$

Equation 2.29 describes the input gyroscopes and accelerometers measurements input to the l-frame mechanization and the output curvilinear coordinates, three velocity component, and three attitude components.

2.4 Estimating the Navigation State

Solving the vector differential equations, through integration, will result in a time variable state vector with kinematic sub-vectors for position, velocity, and attitude. In the literature, the integration algorithms are often called the mechanization equations. This term obviously dates back to the time when stable platform systems were the norm and a specific platform orientation was actually mechanized, as for instance in the local-level system or the space-stable system. The accelerometers mounted on the platform were isolated from rotational vehicle dynamics and the computation of the transformation matrix was replaced by a system of gimbals, supplemented by platform commands. Although the algorithms used for strapdown inertial systems can be considered as an analytical form of the platform mechanization, we will use the term integration equations in the following to distinguish between stable platform and strapdown systems.

The integration algorithm can be summarized using Figure 2.4

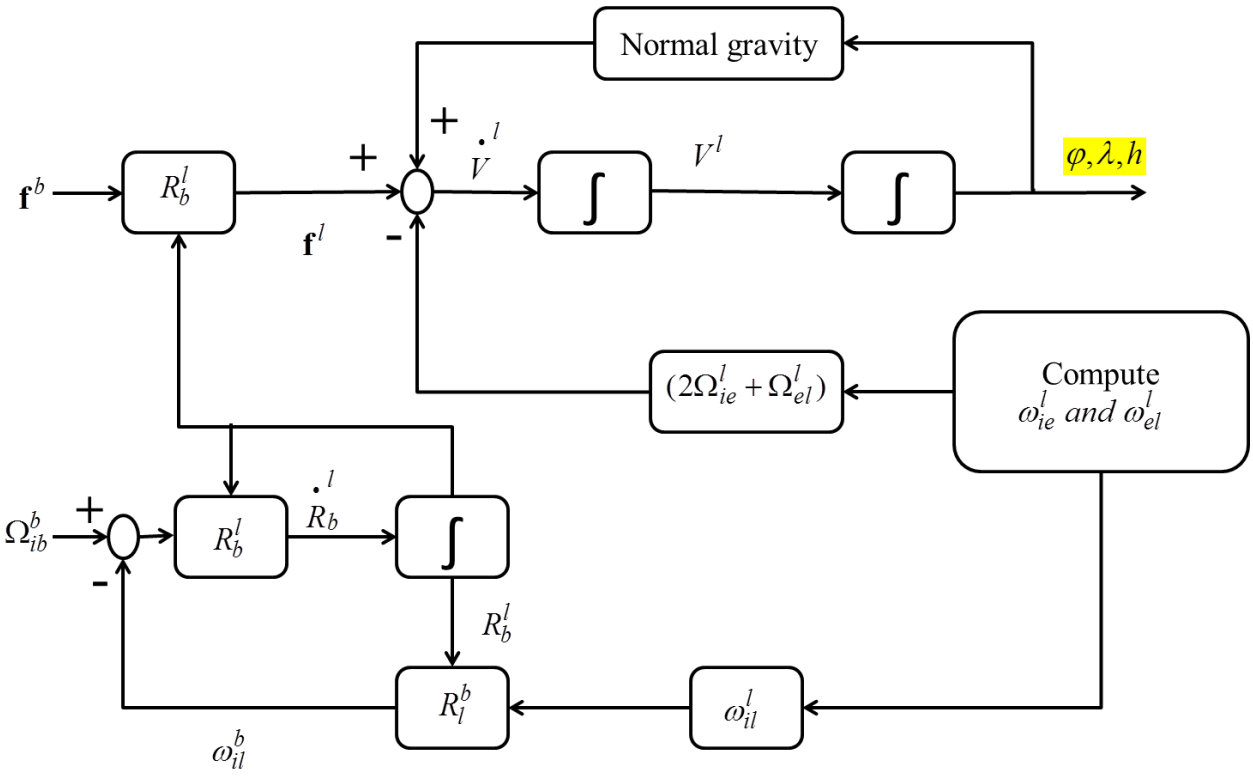


Figure 2.4: Mechanization of the aircraft motion in l-frame

The following table is a summary of the required parameters to implement the mechanization algorithm shown in Figure 2.4 for aircraft motion.

Table 2-1: Summary of the required parameters for l-frame mechanization

Parameter	Computed from
$\mathbf{f}^b$	Accelerometers
Ω_{ib}^b	Gyroscopes
D^{-1}	Reference ellipsoid and lattiutde
G^l	Gravity model
M	Known
N	Known

R_b^l	Alignment
ω^e	Known

2.4.1 Input measurements

The input to computation process of the mechanization algorithm is:

1. Scaled and compensated angular increments $\tilde{\theta}_{ib}^{i}$ computed from the body-sensed angular rates.
2. Velocity increments $\Delta \tilde{v}_f^b$ computed from the body-sensed specific force.

Given that the gyroscope drift g^{drift} and the accelerometer bias a^{bias} are computed from the calibration process, the raw measurements are corrected as the following:

$$\begin{aligned} \theta_{ib}^b &= \tilde{\theta}_{ib}^b - g^{drift}\Delta t \\ \Delta v_f^b &= \Delta \tilde{v}_f^b - a^{bias}\Delta t \end{aligned} \qquad \textbf{2.30}$$

2.4.2 Angular increment in the l-frames θ_{lb}^b

The aircraft angular rates with respect to the l-frame are given as:

$$\omega_{lb}^b = \omega_{ib}^b - \omega_{il}^b = \omega_{ib}^b - R_l^b \omega_{ie}^e \qquad \textbf{2.31}$$

Where:

ω_{ib}^b are the angular rates sensed by the gyroscopes.

ω_{ie}^e is the earth rotation rate.

R_l^b is the transpose of the rotation matrix R_b^l.

Using the obtained roll, pitch and azimuth angles, the rotation matrix from body to l-frame is constructed:

$$R_b^l = R_3(\psi)R_2(\phi)\,R_1(\theta) = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \qquad \textbf{2.32}$$

where:

$$R_3(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \; R_2(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix} \; R_1(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

$\phi, \theta, and\ \psi$ are the roll, pitch, and azimuth angels respectively.

The earth rate in the b-frame ω_{il}^b in equation can be expressed as:

$$\begin{aligned} \omega_{il}^b &= R_l^b \omega_{il}^l = R_l^b (\omega_{ie}^l + \omega_{el}^l) \\ &= R_\ell^b \left[\begin{pmatrix} 0 \\ \omega^e \cos\phi \\ \omega^e \sin\phi \end{pmatrix} + \begin{pmatrix} \dfrac{-V^n}{M+h} \\ \dfrac{V^e}{N+h} \\ \dfrac{V^e \tan\phi}{N+h} \end{pmatrix} \right] = R_\ell^b \begin{pmatrix} \dfrac{-V^n}{M+h} \\ \dfrac{V^e}{N+h} + \omega^e \cos\phi \\ \dfrac{V^e \tan\phi}{N+h} + \omega^e \sin\phi \end{pmatrix} \end{aligned} \qquad \textbf{2.33}$$

The angular increments for the Earth rotation in the body frame are computed by integrating equation 2.33 as follows:

$$\theta_{il}^{b} = \omega_{il}^{b}\Delta t = \begin{pmatrix} d\theta_x^b \\ d\theta_y^b \\ d\theta_z^b \end{pmatrix} \qquad \textbf{2.34}$$

The angular increments of the body motion with respect to the l-frame are computed by integrating equation 2.31 as follows:

$$\begin{aligned} \theta_{lb}^{b} &= \theta_{ib}^{b} - \theta_{il}^{b} \\ \begin{pmatrix} \theta_x^b \\ \theta_y^b \\ \theta_z^b \end{pmatrix} &= \begin{pmatrix} \Delta\theta_x^b \\ \Delta\theta_y^b \\ \Delta\theta_z^b \end{pmatrix} - \begin{pmatrix} d\theta_x^b \\ d\theta_y^b \\ d\theta_z^b \end{pmatrix} \end{aligned} \qquad \textbf{2.35}$$

The first term in the right hand side of equation 2.35 is computed from the gyroscope measurements and the second term from the Earth rotation rate and the rotation matrix R_l^b .

2.4.3 Rotation matrix update

The rotation matrix R_b^l is updated by following the set of differential equations shown in equation 2.27. The Quaternion approach is used in the update because it deals with the singularity problems of the Euler angles at the 90^o angle. The quaternion is a 4 elements vector represented in space and contains the amplitude in one element and the direction is described using the three remaining elements.

The quaternion update can be described in terms of the angular increments in the l-frame θ_{lb}^{b} as:

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}_{(k+1)} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}_{(k)} + 1/2 \begin{pmatrix} c & s\theta_z^b & -s\theta_y^b & s\theta_x^b \\ -s\theta_z^b & c & s\theta_x^b & s\theta_y^b \\ s\theta_y^b & -s\theta_x^b & c & s\theta_z^b \\ -s\theta_x^b & -s\theta_y^b & -s\theta_z^b & c \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}_{(k)} \qquad \textbf{2.36}$$

Where:

The angular increments $(\theta_x^b, \theta_y^b, \theta_z^b)$ are computed from equation 2.49.

The incremental angle of the body rotation θ is computed using $\theta^2 = (\theta_x^b)^2 + (\theta_y^b)^2 + (\theta_z^b)^2$.

The coefficients *c* and *s* are computed as:

$$c = 2(\cos\tfrac{\theta}{2} - 1) = -\tfrac{1}{4}\theta^2 + \tfrac{1}{192}\theta^4 + \cdots$$
$$s = \tfrac{2}{\theta}\sin\theta = 1 - \tfrac{1}{24}\theta^2 + \tfrac{1}{1920}\theta^4 + \cdots$$

Therefore, the rotation matrix from the b-frame to the l-frame R_b^l is updated as the following:

$$R_b^l = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} = \begin{pmatrix} (q_1^2 - q_2^2 - q_3^2 + q_4^2) & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & (q_1^2 - q_2^2 - q_3^2 + q_4^2) & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & (q_1^2 - q_2^2 - q_3^2 + q_4^2) \end{pmatrix} \quad \textbf{2.37}$$

2.4.4 The specific force transformation to the l-frame

The specific force sensed by the accelerometer is transformed to the l-frame by:

$$\mathbf{f}^l = R_b^l \mathbf{f}^b \quad \textbf{2.38}$$

The strapdown accelerometers output is computed by integrating the specific force components over a small sampling period. Therefore, the velocity increments Δv_f^l are computed as the following:

$$\Delta v_f^l = \int_{t_k}^{t_{k+1}} R_b^l(t)\mathbf{f}^b(t)dt \quad \textbf{2.39}$$

Since the rotation matrix R_b^l is varying with time, it can be expressed in first order approximation as:

$$R_b^l(t) = R_b^l(t_k)(\mathrm{I} + S^b) \qquad \textbf{2.40}$$

Where S is the skew symmetric form of the small incremental changes of rotation over the time epoch $[t_k, t]$. The matrix $(\mathrm{I} + S^b)$ is expressed as the orthogonal transformation between the b-frame at time t_k and the b-frame at time *t*, i.e.

$$R_b^l(t_k, t) \cong \mathrm{I} + S^b = \begin{pmatrix} 1 & -\theta_z^b & \theta_y^b \\ \theta_z^b & 1 & -\theta_x^b \\ -\theta_y^b & \theta_x^b & 1 \end{pmatrix} \qquad \textbf{2.41}$$

Assuming constant angular rate and linear acceleration over small time epochs, the total velocity increments Δv_f^l is computed by substituting equation 2.40 in equation 2.39 as the following:

$$\begin{aligned} \Delta v_f^l &= R_b^l(t_k)(\mathrm{I} + \tfrac{1}{2} S^b)\Delta v^b \\ &\textit{or} \\ \Delta v_f^l &= R_b^l(t_{k+1})(\mathrm{I} - \tfrac{1}{2} S^b)\Delta v^b \end{aligned} \qquad \textbf{2.42}$$

Where $R_b^l(t_k)$ and $R_b^l(t_{k+1})$ are the transformation matrix updated at time epoch (*k*) and (*k+1*) respectively.

2.4.5 Velocity update

The velocity update is computed by applying the Coriolis and gravity correction as follows:

$$\Delta v^l = \Delta v_f^l - (2\omega_{ie}^l + \omega_{el}^l) \times v^l \Delta t + \zeta^l \Delta t \qquad \textbf{2.43}$$

Where:

$\zeta^l = (0 \quad 0 \quad \zeta)^T$, and ζ is the normal gravity at the geodetic latitude φ and ellipsoidal height *h* given as (Heiskanen and Moritz 1967):

$$\zeta = a_1(1 + a_2 \sin^2 \varphi + a_3 \sin^4 \varphi) + (a_4 + a_5 \sin^2 \varphi)h + a_6 h^2 \qquad \textbf{2.44}$$

Where:

$a_1 = 9.7803267715$ $\quad$ $a_4 = -0.0000030876910891$
$a_2 = 0.0052790414$ $\quad$ $a_5 = 0.0000000043977311$
$a_3 = 0.0000232718$ $\quad$ $a_6 = 0.0000000000007211$

Using the modified Euler formula, the velocity at the current epoch is computed as the following:

$$V^{\ell}(t_{k+1}) = V^{\ell}(t_k) + \frac{1}{2}\left(\Delta V^{\ell}(t_k) + \Delta V^{\ell}(t_{k+1})\right) \qquad \textbf{2.45}$$

Where:

$$V^{\ell} = \begin{pmatrix} V^e & V^n & V^u \end{pmatrix}^T$$

2.4.6 Position update

The position update is computed using the same approach for the velocity update.. The height component is directly mapped to the vertical velocity and it is updated using the following equation:

$$h(t_{k+1}) = h(t_k) + \frac{1}{2}\left(V^u(t_{k+1}) + V^u(t_k)\right)\Delta t \qquad \textbf{2.46}$$

The Latitude and Longitude components are not directly mapped to velocity components but calculated through the previously described equation 2.29 such that:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{R+h} & 0 \\ \frac{1}{(R+h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V^e \\ V^n \\ V^u \end{pmatrix} \qquad \textbf{2.47}$$

And using the same update trapezoidal rule they are calculated through the following equations:

$$\varphi(t_{k+1}) = \varphi(t_k) + \frac{1}{2}\frac{\left(V^n(t_{k+1}) + V^n(t_k)\right)}{R+h}\Delta t \qquad \textbf{2.48}$$

$$\lambda(t_{k+1}) = \lambda(t_k) + \frac{1}{2}\frac{\left(V^e(t_{k+1}) + V^e(t_k)\right)}{(R+h)\cos\varphi}\Delta t \qquad \textbf{2.49}$$

2.4.7 Attitude update

Using equation 2.37 for R_b^l, the attitude angles roll, pitch, and azimuth updates are computed as:

$$\begin{aligned} \theta &= \tan^{-1}\left(\frac{R_{32}}{R_{33}}\right) \\ \phi &= \sin^{-1}\left(R_{31}\right) \\ \psi &= \tan^{-1}\left(-\frac{R_{21}}{R_{11}}\right) \end{aligned} \qquad \textbf{2.50}$$

Chapter Three: **SPEEDED UP ROBUST FEATURES (SURF) ALGORITHM**

This chapter reviews the concepts and background of image matching algorithms which play a key role in the VBN. Because the UAV images can be oblique and at different scale from the database images, a robust scale and rotation invariant image matching algorithm is important for VBN of aerial vehicles. The chapter starts with the most important research work done in image matching along with the corresponding techniques used. Next, the modified SURF algorithm will be explained starting from the interest point detection to modified SURF descriptors building and finally implementation of the modified SURF algorithm using Graphics Processing Unit (GPU) Compute Unified Device Architecture (CUDA) to meet the real-time processing requirements, while keeping the high accuracy, which is is for the VBN real-time operations (Sheta et al. 2012e).

3.1 Introduction

Image matching algorithms play a basic role in many applications such as surveillance, law enforcement, aerial mapping, and navigation of unmanned land and aerial vehicles. Comparing images pixel by pixel (named as correlation) is one of the classical image matching technqiues (Johnson and Kotz 1997). However, this approach suffers from the high computation load and may result in wrong result due to scale and rotation variations in the matched images. To overcome the high computational load for the classic correlation, the concept of Bounded Partial Correlation (BPC) has been introduced (Di Stefano et al. 2005). In this approach, two sufficient conditions based on Cauchy-Schwarz inequality at each image position are introduced such that the computational load is reduced. Using Cauchy-Schwarz inequality, the upper bound of the correlation score is computed at each point instead of computing the correlation process, which

leads to lower computational load. However, it didn't overcome the rotation and scale variation between the matched images. Block matching approach is another method that overcome the computational load problem through parallel processing of blocks (Alkaabi and Deravi 2005; Mattoccia et al. 2007). However, the Block matching approach did not cover the scale invariance problem. To address the texture features problem, (Jalil et al. 2006) introduced the Principal Component Analysis (PCA) and wavelet transform to overcome the rotation invariance challenge in texture feature problem. This approach estimates the principal component direction by the PCA and rotates the original image to the detected angle by the PCA. Then, the feature extraction is conducted by the wavelet transform. However, the use of wavelet transform in feature extraction doesn't provide scale and intensity invariance.

(Borgefors 1988) introduced the multiresolution matching approach. This approach reduced the computation load through the use of lower resolution images. However, experimental results indicates that the computation time achieved in the proposed approach (was 8 minutes for image size of 512 x 512 pixels) is too slow to be employed in real-time applications such as VBN. (Fitch et al. 2005) introduced image matching in the frequency domain. In this approach, the number of operations for matching surface generation computational load was reduced by a factor of 820 for image size of 512×512 pixels. To continue with image matching in frequency domain, (Essannouni et al. 2007) introduced L4 norm as an approach for computationally efficient approach for image matching. However, the proposed image matching in the frequency domain has a disadvantage coming from the overhead resulting from transforming back to the spatial domain which affects the real-time requirements for VBN.

(Berg et al. 2005; Makarov 2006) introduced the shape matching approach. In this approach, descriptors are based on geometric blur point which calculates a cost function for the similarity

of point descriptors and the geometric distortion. (El Oirrak et al. 2002) investigated the same approach but through the use of Fourier descriptors to describe the shape features.

In 2004, (Lowe 2004) introduced Scale Invariant Feature Transform (SIFT) which outperform other approaches for interest point detection and matching. SIFT approach is a rotation and scale invariant which is a crucial point to be considered when dealing with image matching algorithms. In addition, it is considered as relatively fast approach for real-time application. However, when (Lowe 2004) tried to speed up the algorithm performance in the matching step through the best-bin-first approach, more mismatches appeared which affect the matching process.

In this thesis, matches between the geo-referenced database images and those captured in real-time are found by employing the fast SURF algorithm. SURF, sometimes referred to as the Fast-Hessian detector, is essentially based on the Hessian matrix with Laplacian based detectors such as Difference of Gaussian (DoG) (Bay et al. 2006). SURF descriptors describes the gradient information in the point of interest neighbourhood through Haar wavelet responses (Anqi and Dudek 2010).

The algorithm consists mainly of two steps: the first is the detection of points of interest and the second is the creation of descriptors for each point. The integral image approach is used to improve the performance of the algorithm computational time prospective. The block diagram for the SURF algorithm is shown in Figure 3.1.

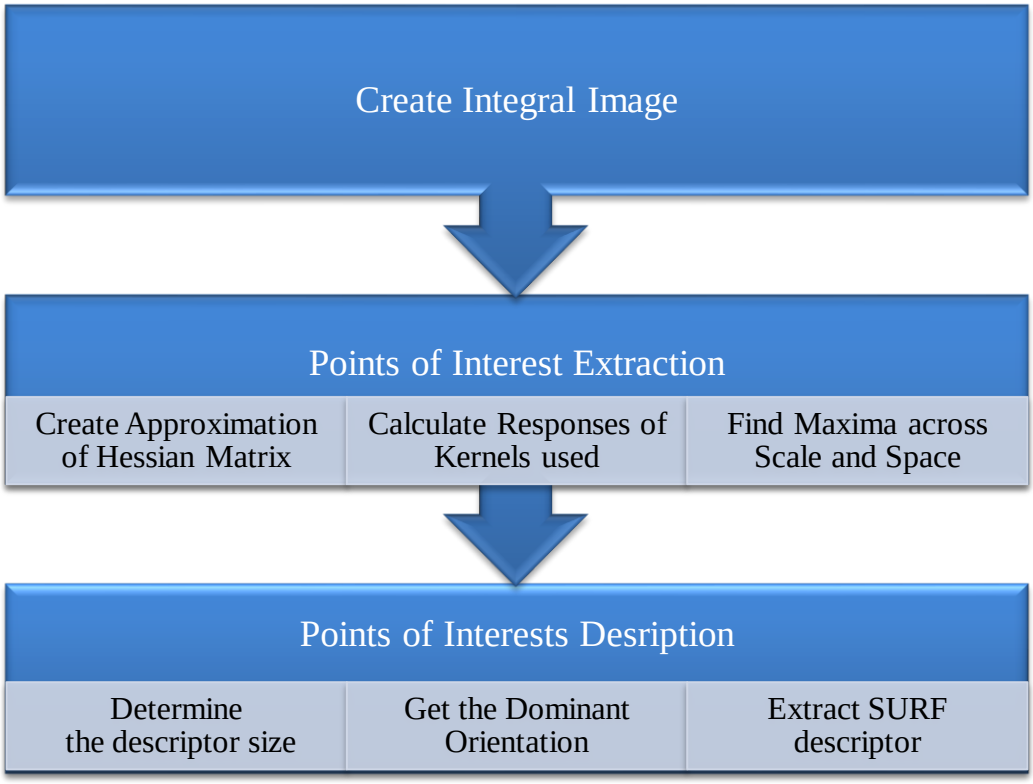


Figure 3.1: SURF block diagram (Sheta et al. 2012b)

3.2 Interest point detection

Interest point detection includes three processes. The first process is integral image computation for fast implementation of the proposed methodology. The second process is Hessian detectors computation which includes creating Hessian based scale space pyramid to find the interest point candidates. Finally, the interest point is detected using 3D Non-Maximum Suppression algorithm for global interest point detection.

To achieve fast robust features, the SURF algorithm employs the integral images approach which reduces the computation time.

3.2.1 Integral images

Occasionally, this approach is referred to the summed area table (Kruis 2010) and is based on forming an integral image $I_{\Sigma}(x)$ from the summing of pixels' intensities of the input image I within a rectangular region formed around location x as follows (Bay et al. 2006):

$$I_{\Sigma}(x) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i, j) \qquad \textbf{3.1}$$

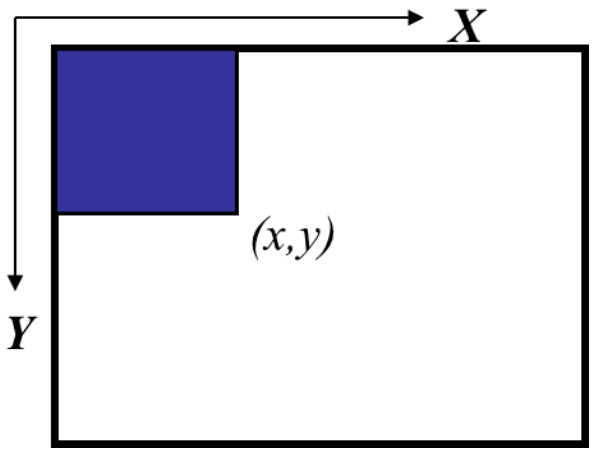


Figure 3.2 : Integral image basic idea

The integral image computes a value at each pixel (*x*,*y*) that is the sum of the pixel values above and to the left of (*x*,*y*) as shown in Figure 3.2. With the recursive definition shown below, the integral image can be computed quickly in one pass through the image as shown in Equation 3.2 and Figure 3.3.

$$\begin{aligned} sum(x, y) &= sum(x, y-1) + i(x, y) \\ I(x, y) &= I(x-1, y) + s(x, y) \end{aligned} \qquad \textbf{3.2}$$

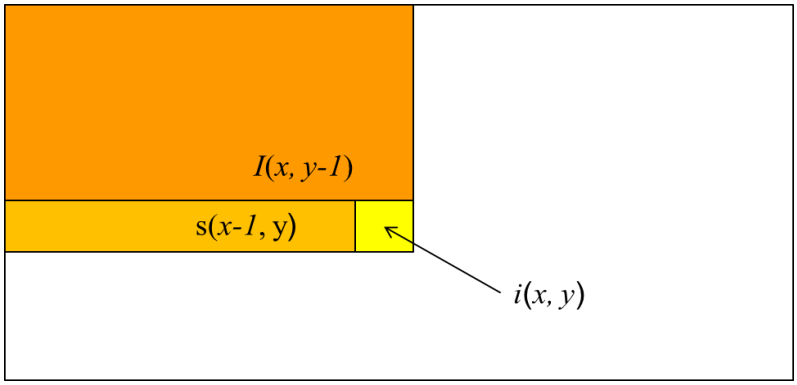


Figure 3.3 Recurrsive definition for integral image

The integral image utilizes three algebraic operations to compute the summation of the intensities in the sub-region of the image as shown in Figure 3.4. The summation of the pixels within rectangle 4 is computed with four array references. The value of the integral image at location A is the sum of the pixels in rectangle 1. The value at location B is 1+2, at location C is 1+3, and at location D is 1+2+3+4. The summation within rectangle 4 is computed as D+A-(B+C).

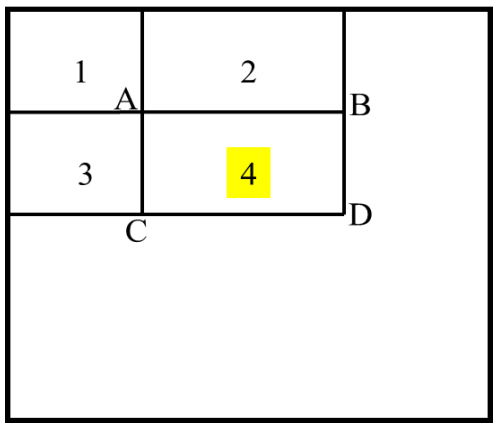


Figure 3.4: The summation of the pixels within rectangle 4

3.2.2 Hessian detectors

A Hessian matrix can be used as a good detector because of its high performance in computational time and accuracy. Scale selection can be achieved through the determinant of the Hessian (Bay et al. 2006) or Hessian – Laplace detector (Mikolajczyk and Schmid 2001).

The Hessian matrix $H(x,\sigma)$ at a given point $x=(x,y)$ in an image I where x at scale σ is defined as:

$$\mathcal{H}(x,y)=\begin{pmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{pmatrix} \qquad \textbf{3.3}$$

Where $L_{xx}(x,\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with image I in point x and similarly for $L_{xy}(x,\sigma)$ and $L_{yy}(x,\sigma)$ (Bay et al. 2006).

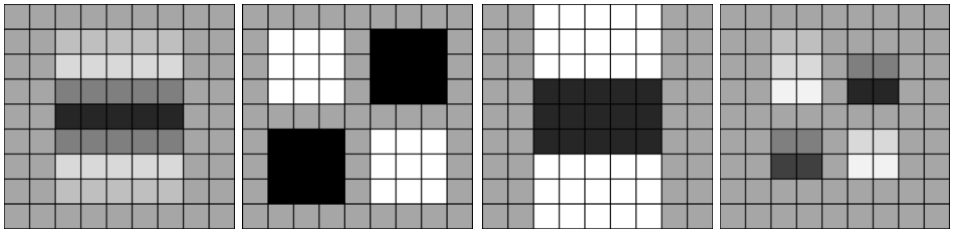


Figure 3.5: Discretized and cropped Gaussian and Box filter approximation for interest point detection (Bay et al. 2006)

Figure 3.5 shows, from left to right, the Gaussian second order partial derivative in y direction (L_{yy}), the *xy* direction (L_{xy}) and the box filter approximation utilized in the SURF algorithm in the *y* direction (D_{yy}) and the *xy* direction (D_{xy}).

The box filter approximation was inspired by Scale Invariant Feature Transform (SIFT)'s success with the Laplacian of Gaussian (LOG). The Hessian matrix approximation can be expressed as:

$$\det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \qquad \textbf{3.4}$$

Where w is the relative weight of the filter response and is given by the following formula for a 9×9 box filter and σ = 1.2 (Bay et al. 2006):

$$w = \frac{\left|L_{xy}(1.2)\right|_F \left|D_{yy}(9)\right|_F}{\left|L_{yy}(1.2)\right|_F \left|D_{xy}(9)\right|_F} \qquad \textbf{3.5}$$

3.2.3 Scale space representation

Scale space representation is defined as the convolution of a given image *f(x,y)* with a Gaussian kernel (Morita 1997):

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma} e^{-(x^2 + y^2)/2\sigma} \qquad \textbf{3.6}$$

Such that the resulting signal $L(x, y; \sigma) = g(x, y; \sigma) * f(x, y)$ is a coarser scaled representation of the original signal.

When dealing with images, scale space representation is implemented as an image pyramid, as shown in Figure 3.6. In this representation, images are smoothed with Gaussian kernels and subsampled so that a higher level of the pyramids is achieved.

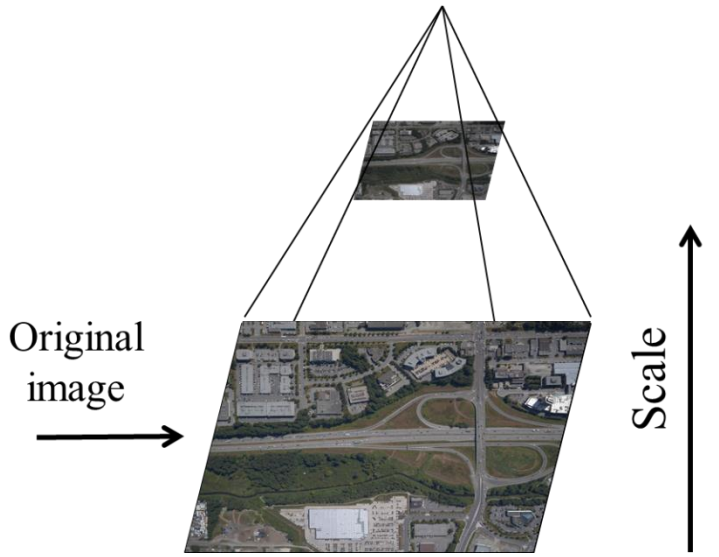


Figure 3.6: Image pyramid for scale space representation of an image (Sheta et al. 2012b)

Interest points must be localized at different scales. As shown in (Lowe 2004), the SIFT approach uses Difference of Gaussians (DoG), where the pyramid layers are subtracted, to find the edges and blobs. However, in the SURF approach the scale space representation is achieved through up-scaling the filter size rather than changing the image size through the image pyramids.

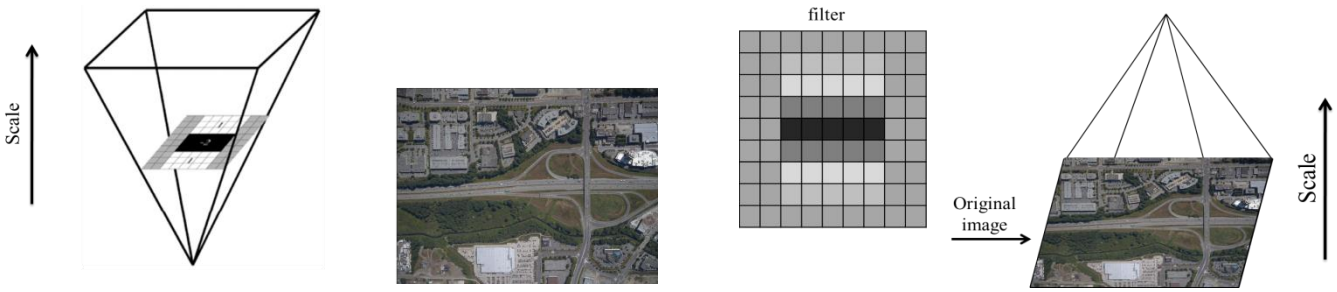


Figure 3.7: SURF implementation for scale space representation to the left where SIFT implementation is shown to the right (Sheta et al. 2012b)

As shown in Figure 3.7, the advantage to using box filters and integral image principles is the high computational efficiency for the SURF approach compared to the SIFT approach since we only change the box filter size in the SURF approach while changing the image size and applying the filter to each image size in the image pyramid in the SIFT approach.

In the SURF approach, the box filter starts off with a 9×9 size filter as the initial scale layer where it is referred as scale $s=1.2$ (the approximated Gaussian derivative with $\sigma=1.2$) and instead of having image pyramids, the original image will be filtered by larger masks. The scale space domain is represented by octaves which can be defined as the filter responses resulting from convolution of the original image with increased size filters.

The first filter used in the scale space representation is of a 9×9 size. Through this filter, the blob response of the image for the smallest scale is calculated. To change the filter size between two successive scales, an increase of 2 pixels (one pixel at each side) is necessary such that the size of the filter is kept uneven. This yields an increased filter size with 6 pixels as shown in Figure 3.8.

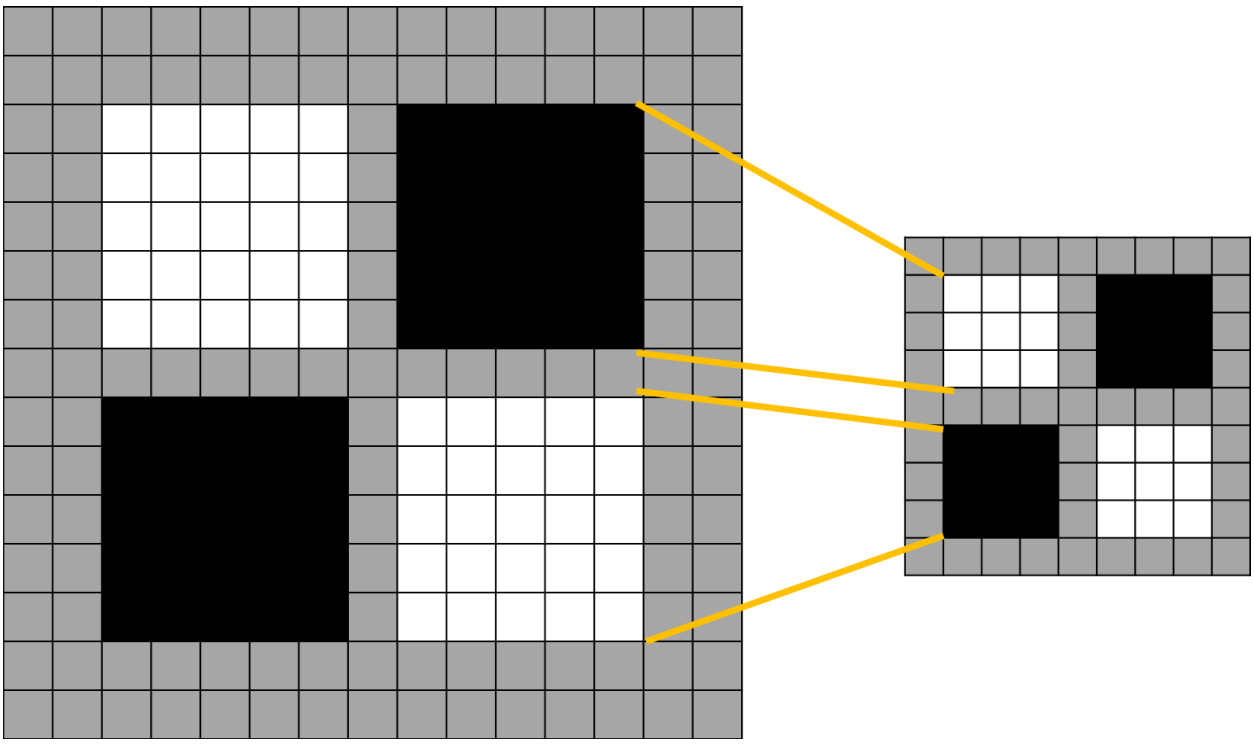


Figure 3.8: Filters D_{xy} for two successive scale levels (9×9 and 15×15) (Bay et al. 2008)

As mentioned earlier, the first filter size used for blob detection is 9×9 for the first octave and then filters of varying sizes, more specifically 15×15, 21×21, and 27×27, are applied to the image.

3.2.4 3D non-maximum suppression for interest point localization

Applying non-maximum suppression to a 3×3×3 neighbourhood localizes the interest points in the image over different scales as shown in Figure 3.9. In this figure, interest point localization is established both spatially and over the neighbouring scales of the pixels.

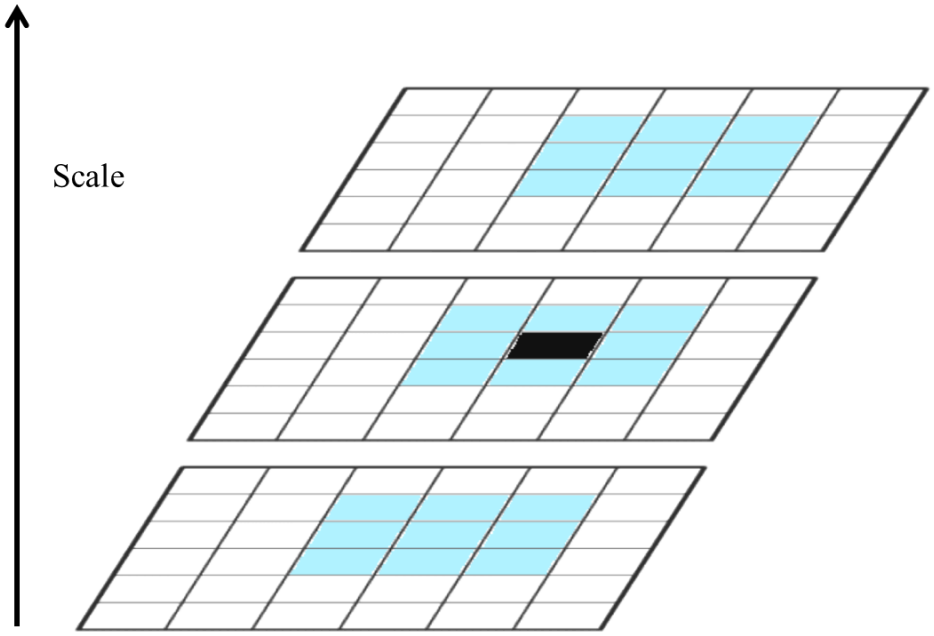


Figure 3.9: 3D non-maximum suppression concept for interest point localization (Sheta et al. 2012b)

Figure 3.9 shows that Non Maximum Suppression (NMS) can be defined as a process in which a candidate interest point (central dark pixel) is considered as an interest point if the intensities of the pixels around it are smaller than the intensity value of the candidate interest point within a certain neighbourhood around it.

The neighbourhood around the interest point can be expressed as follows: for the 1D case, given M pixels to the left and right of the interest point, the neighbourhood is 2M+1. Consequently, in 3D case, the neighborhood is expressed as a cubic region (2M+1) × (2M+1) × (2M+1) centered on the interest point.

Interpolation of the determinant of the Hessian matrix in scale and image space is then employed as discussed in (Brown and Lowe 2002). The interpolated location of the interest point is determined by finding the blob responses (denoted as N) of the 3D neighbourhood previously defined. Then, the interest point localization is improved through sub-pixel/sub-scale

interpolation by fitting a 3D quadratic polynomialto the scale space representation as shown in 3.7

$$N(X) = N + \left(\frac{\partial N}{\partial X}\right)^T X + \frac{1}{2} X^T \frac{\partial^2 N}{\partial X^2} X \qquad \textbf{3.7}$$

Where $X = \begin{pmatrix} x \\ y \\ s \end{pmatrix}$ is the scale space coordinate and *N(X)* is the determinant of the approximated Hessian matrix (blob response resulting from applying the filter) at point of interest location X.

To determine the maximum of the sub-pixel/sub-scale interest point for this 3D quadratic, the derivative of Equation 3.7 with respect to X is computed and equaled to zero as shown in Equation 3.8 and Figure 3.10.

$$\Delta X = -\left(\frac{\partial^2 N}{\partial X^2}\right)^{-1} \frac{\partial N}{\partial X} \qquad \textbf{3.8}$$

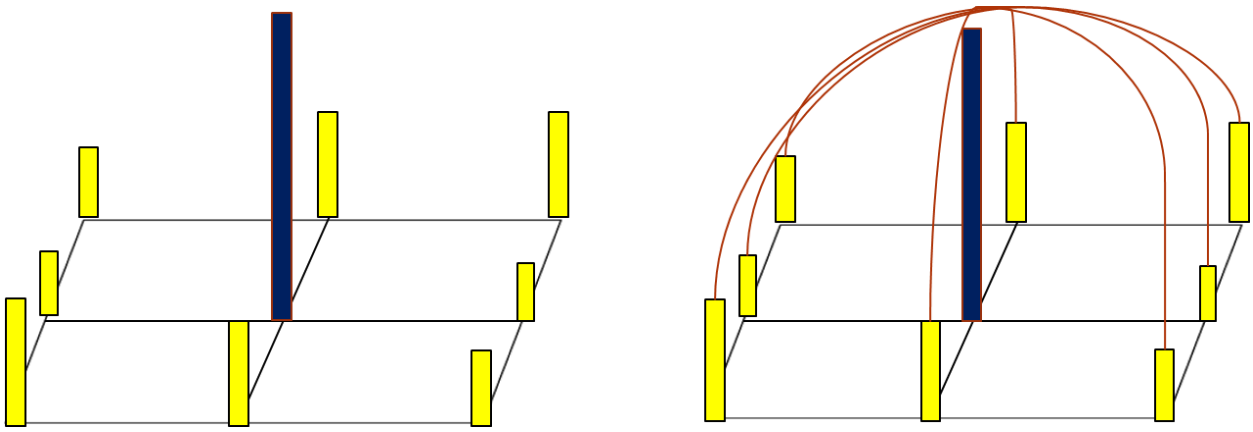


Figure 3.10: 3×3 maximum blob response to the left and parabolic fitting maximum value to the right

3.3 Interest point description and matching

Once the interest point localization has been completed, the interest points must be uniquely described by a descriptor such that the correspondences between two images can be evaluated.

The proposed method is based on the distribution of the blob response within the detected interest point neighborhood.

Based on integral images technique for speed optimization, the blob response within the detected interest point neighborhood is based on the first order Haar wavelet response in *x* and *y* direction. The descriptor dimension can be varied between 36, 64, or 128 depending on the number of sub-regions as will be described later. To achieve fast indexing during the matching process, the sign of the Laplacian is used in the indexing step.

The SURF descriptor is based on two steps. The first step uses the information originating from a circular region around the point of interest which leads to reproducible orientation information (that determines the dominant orientation to help obtain rotation invariant features). Then, the SURF descriptor is extracted from a square region generated and aligned to the selected orientation (Bay et al. 2006).

3.3.1 Interest point orientation assignment

The purpose of interest point orientation assignment is to make the proposed method invariant to image rotation. The Haar wavelet responses are calculated in x and y direction in a circular neighborhood with radius *6s* (this radius represents the neighborhood at which global maxima is detected) around the detected interest point. These wavelet responses are weighted with a Gaussian distribution at the center of the detected interest point and introduced as a horizontal vector along x direction and vertical vector along y direction. The Haar wavelets that were used are shown in Figure 3.11 where dark and light parts represent weights of -1 and 1 respectively. Convolution between these filters and the image represents the image gradient at locations where these filters are applied. Based on the integral image technique, the responses in x and y directions are calculated after six operations.

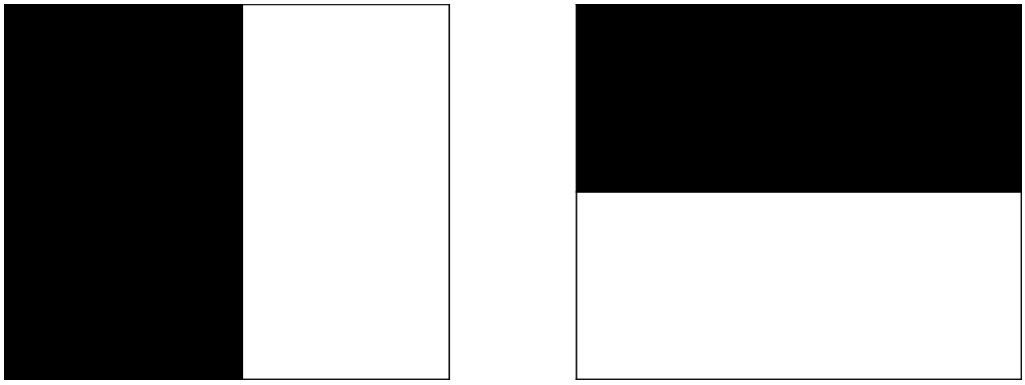


Figure 3.11: Haar wavelet filters used for computing the response in x direction (left) and y direction (right)

A sliding orientation window at angle of $\pi/3$ is employed, as shown in Figure 3.12, to estimate the dominant orientation by calculating the sum of all responses within this window. A new vector is then generated by summing the horizontal and vertical wavelet responses within the window where the longest vector orientation is assigned as the interest point orientation.

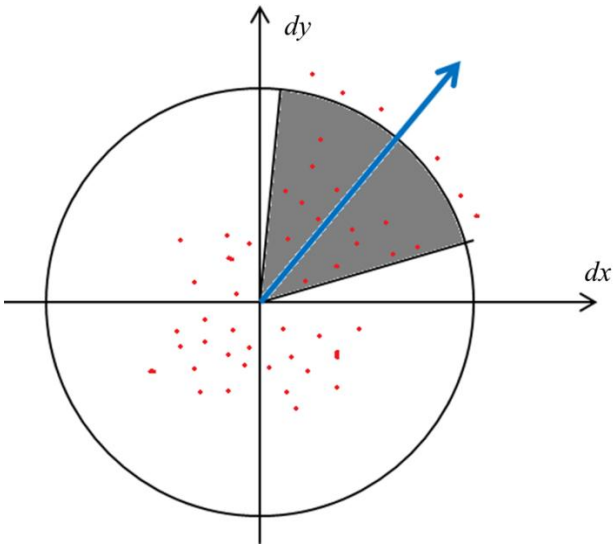


Figure 3.12: Sliding orientation window

3.3.2 Descriptor building

To establish descriptor building, a square region centered on the detected interest point with orientation along the dominant direction, is used. In the case of 64 descriptor length (SURF 64),

the square region is divided into equally 4×4 sub-regions. At each sub-region, a number of sample points are used to compute the corresponding features. The number of sample points that are used affects the accuracy of the matching algorithm. The higher the number of sample points, the better the matching will be.

Tests were done with varying numbers of sub-regions and sample points in each sub-region. SURF 64 Descriptor length varied from 36 (where 3×3 sub-regions are used as shown in Figure 3.13), to 64 (where 3×3 sub-regions are used as shown in Figure 3.14), to 128 (where several of similar features are added to the descriptor as shown in Figure 3.15) are implemented through this thesis.

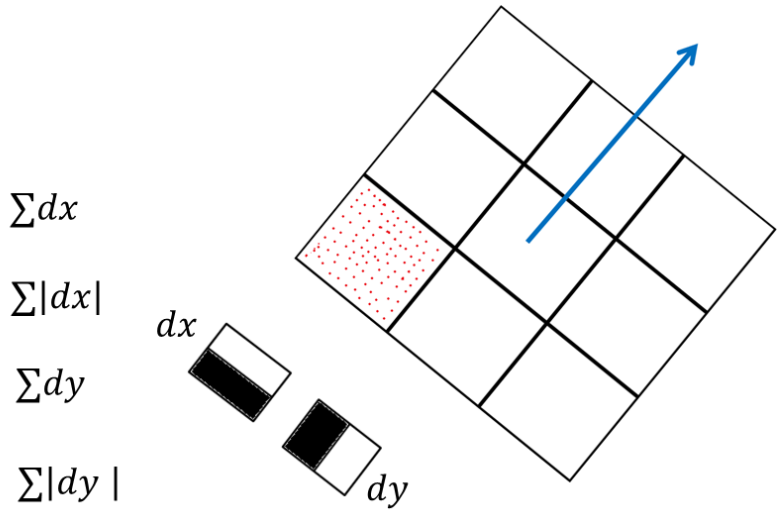


Figure 3.13: Descriptor length 36

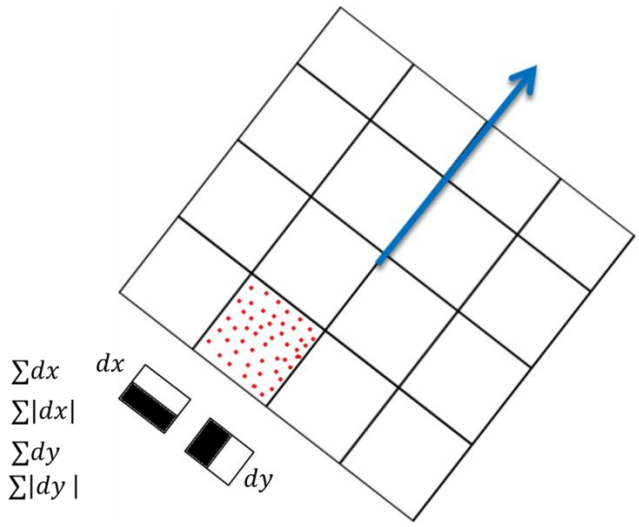


Figure 3.14: Descriptor length 64

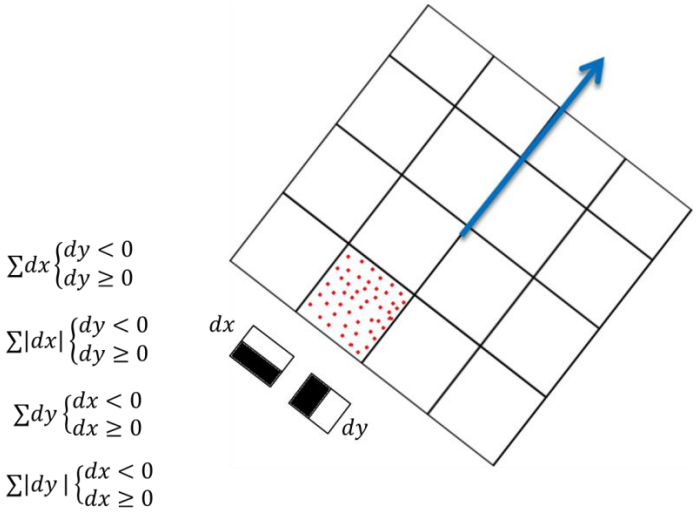


Figure 3.15: Descriptor length 128

For each sub-region, the descriptor vector can be described as *D_V*, where this descriptor vector is four dimensional (in the case of 36 and 64 descriptor length) and presents the intensity structure.

$$D\_V = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right) \qquad \textbf{3.9}$$

This descriptor vector is normalized to achieve invariance to contrast. An example of the effect of this descriptor building on the intensity pattern within a sub-region is shown in Figure 3.16.

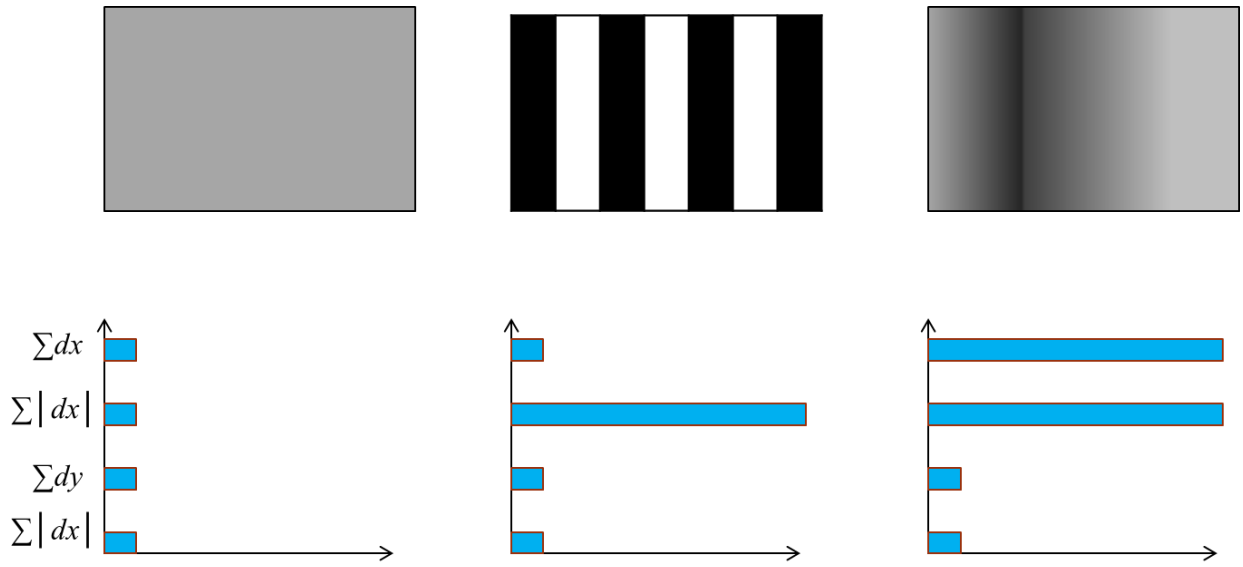


Figure 3.16: The effect of descriptor building on the intensity pattern (Bay et al. 2006)

The implementation of different number of sample points in each sub-region is shown in Figure 3.17.

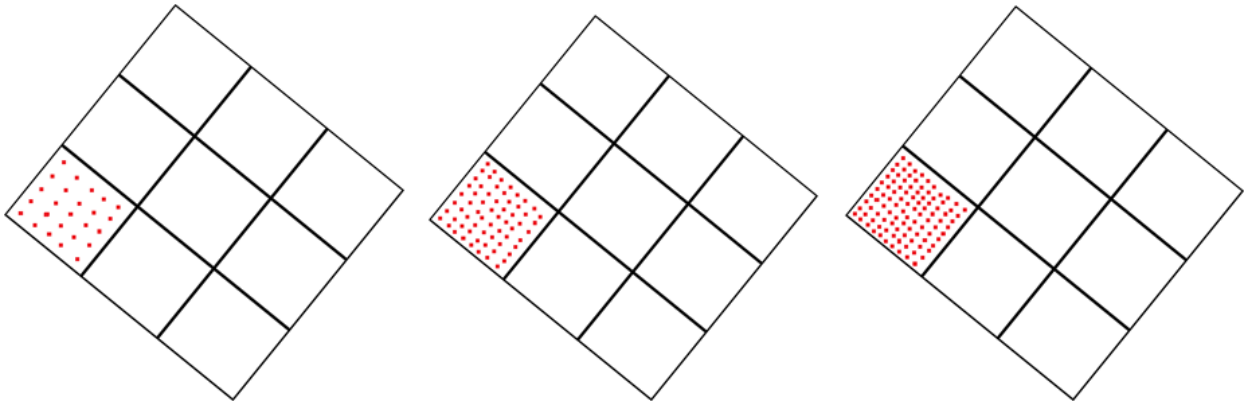


Figure 3.17: Different number of samples in each sub-division from the left 5×5, 8×8, and 10×10 respectively (Sheta et al. 2012b)

3.4 Indexing for correspondence points matching

To achieve fast indexing during the matching process, the sign of the Laplacian is used (Bay et al. 2006). Minimal information is required to increase the speed at which matching occurs between correspondence points without reducing the descriptor performance. This minimal

information is the sign of the Laplacian. To differentiate between bright blob response on dark background and dark blob response in bright background, the sign of the Laplacian is employed. Correspondence points are found in the matching stage when comparing the points with the same type of contrast. As shown in Figure 3.18, the left image represents the traditional way of matching two images, where each interest point is compared to all the other interest points detected in the other image. However, if the information regarding the contrast of the interest point is included (whether it is a dark blob in light background or a light blob in dark background), as it is in the right image of Figure 3.18, then matching will be accomplished with the interest points maintaining the same type of contrast.

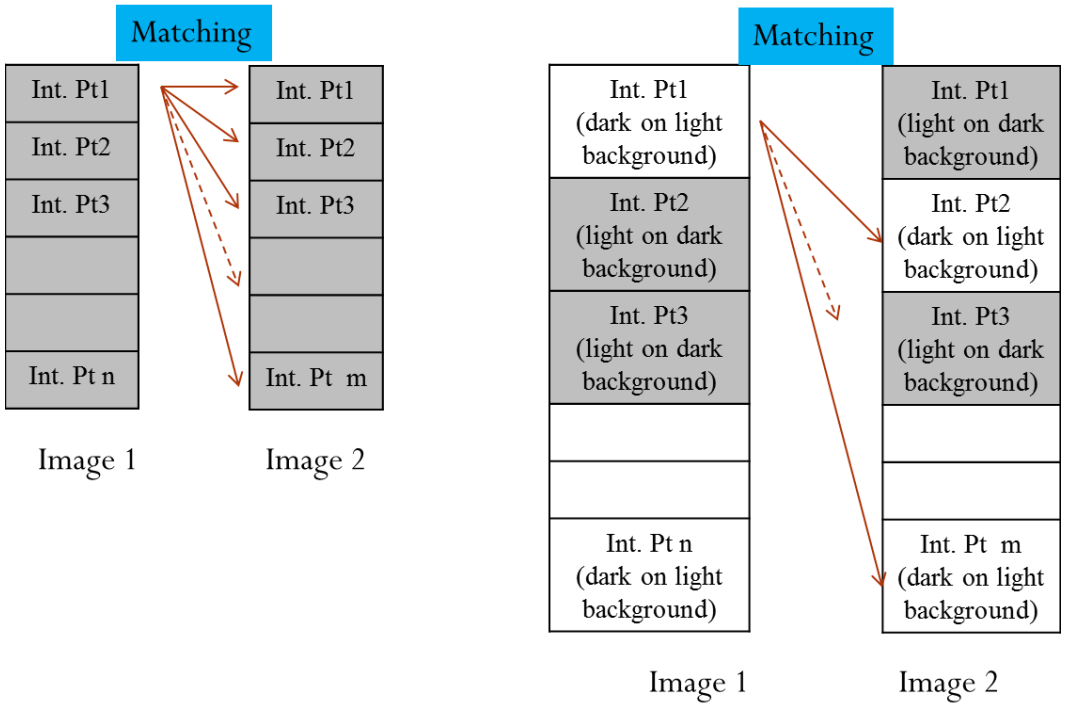


Figure 3.18: Fast indexing based on the sign of the Laplacian (Sheta et al. 2012b)

The matching strategy is based on the Euclidean distance in descriptor space. This approach is referred to as similarity-threshold-based matching strategy.

3.5 Test set and results

To test the developed modified SURF methodology, which include different numbers of descriptor length(SURF-36, SURF-64, and SURF 128) with different number of sample counts in each sub-region an airborne image data set, captured over the Vancouver area, will be used. Performance of the developed methodology against scale and rotation variation is investigated using the repeatability measure. Two data sets have been collected during the test. The first data set was collected during a flight test with vertical camera axis and the second data set was collected during a flight test with oblique camera axis (15°) cross track. The camera specification and the flight information are given in Table 3-1 and Figure 3.19.

Table 3-1: Camera and flight specification

Focal length	50 mm
Pixel size	7.21 μm
CMOS format	24×36 mm (3328×4992 pixels)
Flying speed	100 knots
Flying height	1000 m
Data acquisition rate	≥ 3.5 sec
Tilt angle	15° (roughly)
Area coverage	4km×3km

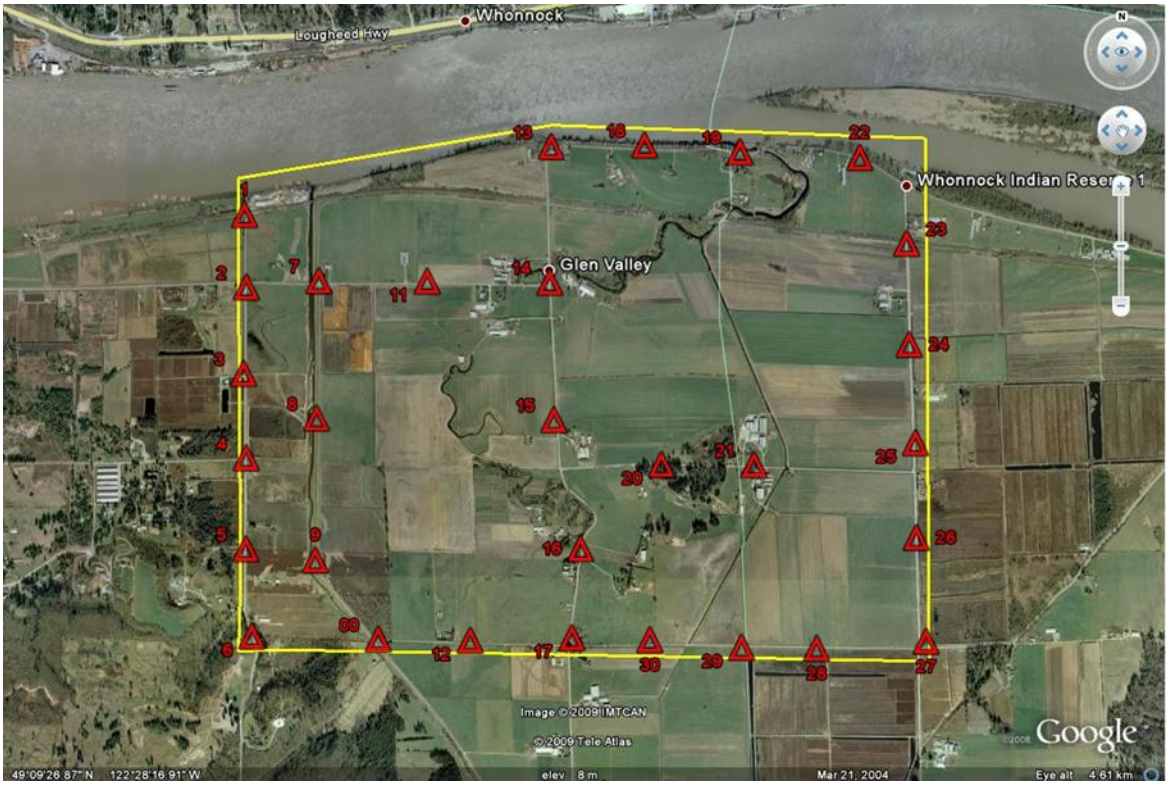


Figure 3.19: Flight area coverage where red triangles represent the GCP used during the test

Tests were conducted using descriptor length (36, 64, and 128) with different number of sample points (5×5, 9×9, and 13×13) in each sub-region. These descriptors were applied to images with different scale and orientation to check the robustness of the developed algorithms.

The repeatability measure is used to provide a measure on the reliability of the developed algorithms for detecting the same interest points under different scale and rotation variations.

Tests were conducted using the original image as a representative of the database and a rotated and scaled image as representation of the real-time images captured from the UAV.

Figure 3.20: Descriptor length 64 with scale variation = 0.2 and rotation = 15 and number of sample points 5x5

Figure 3.21: Descriptor length 64 with scale variation = 0.4 and rotation = 15 and number of sample points 5x5

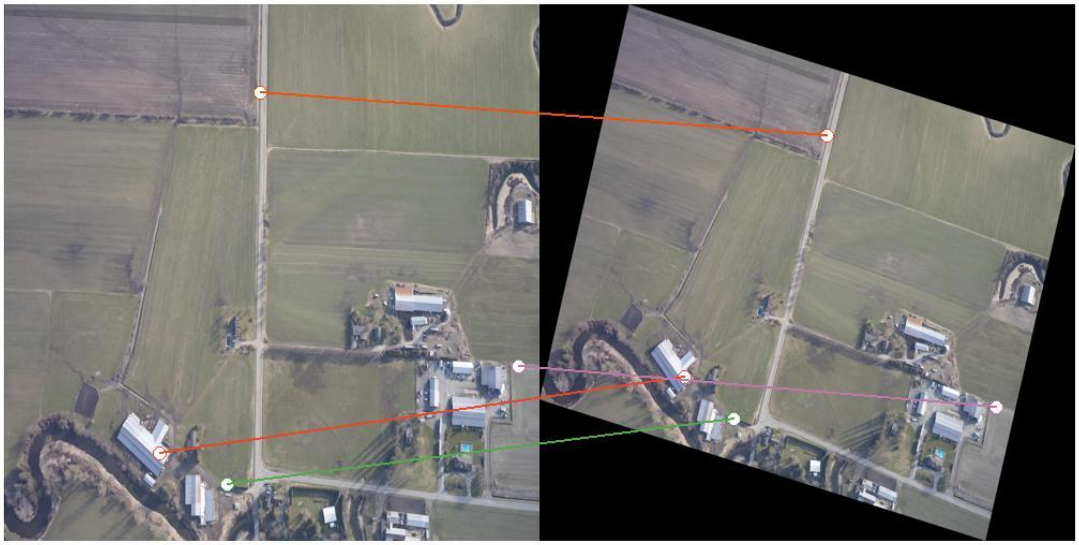


Figure 3.22: Descriptor length 64 with scale variation =0.6 and rotation = 15 and number of sample points 5x5

Figure 3.23: Descriptor length 64 with scale variation = 0.8 and rotation = 15 and number of sample points 5x5

Figure 3.24: Descriptor length 64 with scale variation = 1 and rotation = 15 and number of sample points 5x5

Figure 3.25: Descriptor length 36 with scale variation =0.2 and rotation = 15 and number of sample points 5x5

Figure 3.26: Descriptor length 36 with scale variation =0.4 and rotation = 15 and number of sample points 5x5

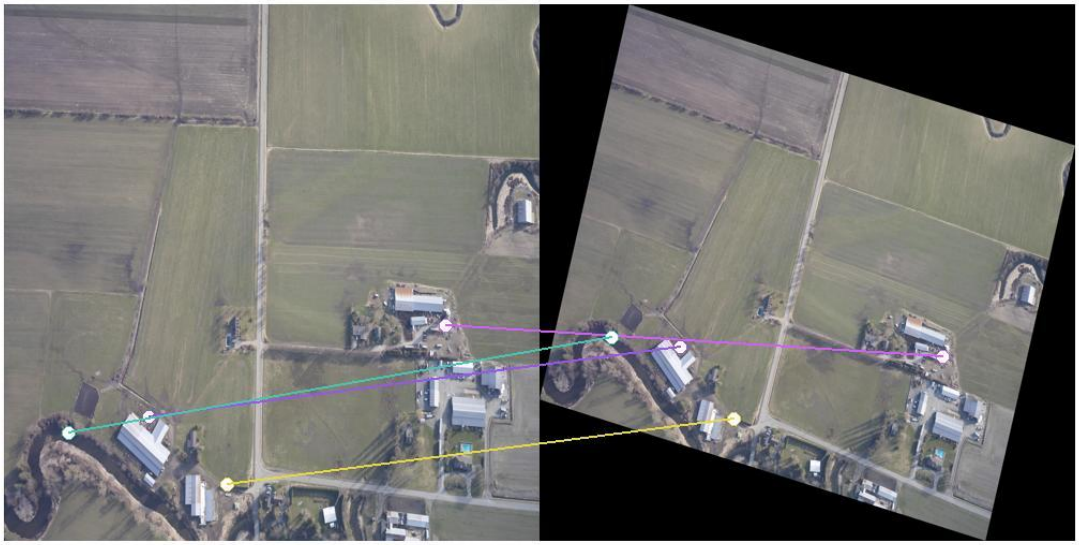


Figure 3.27: Descriptor length 36 with scale variation =0.6 and rotation = 15 and number of sample points 5x5

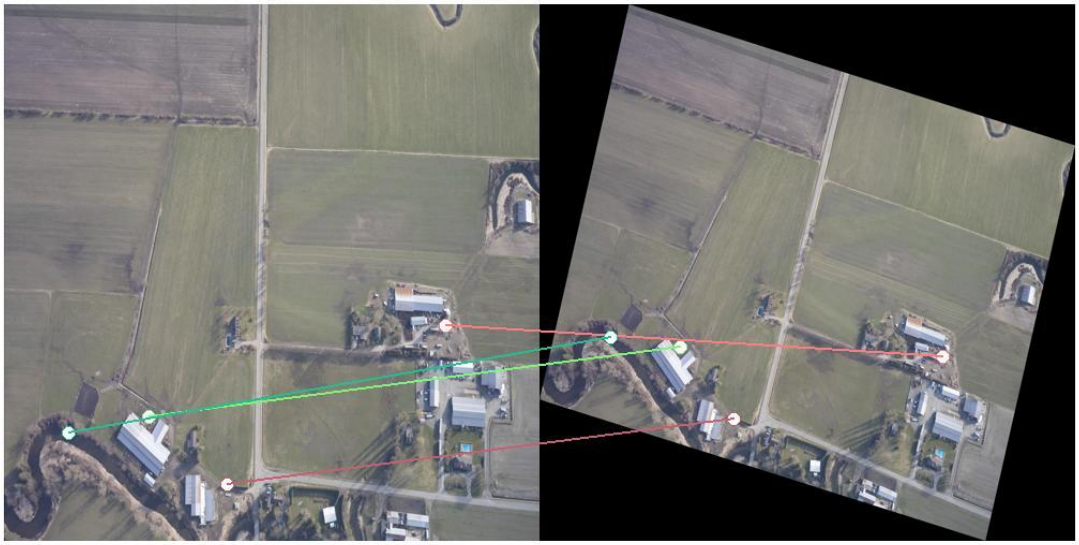


Figure 3.28: Descriptor length 36 with scale variation =0.8 and rotation = 15 and number of sample points 5x5

Figure 3.29: Descriptor length 36 with scale variation =1 and rotation = 15 and number of sample points 5x5

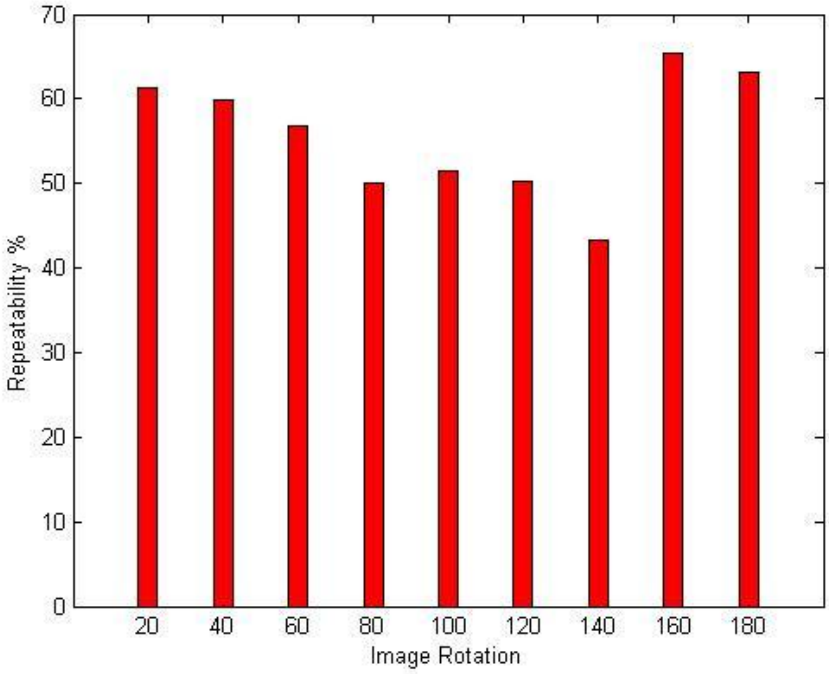


Figure 3.30 : Repeatability measure for descriptor length 64 and scale 0.2

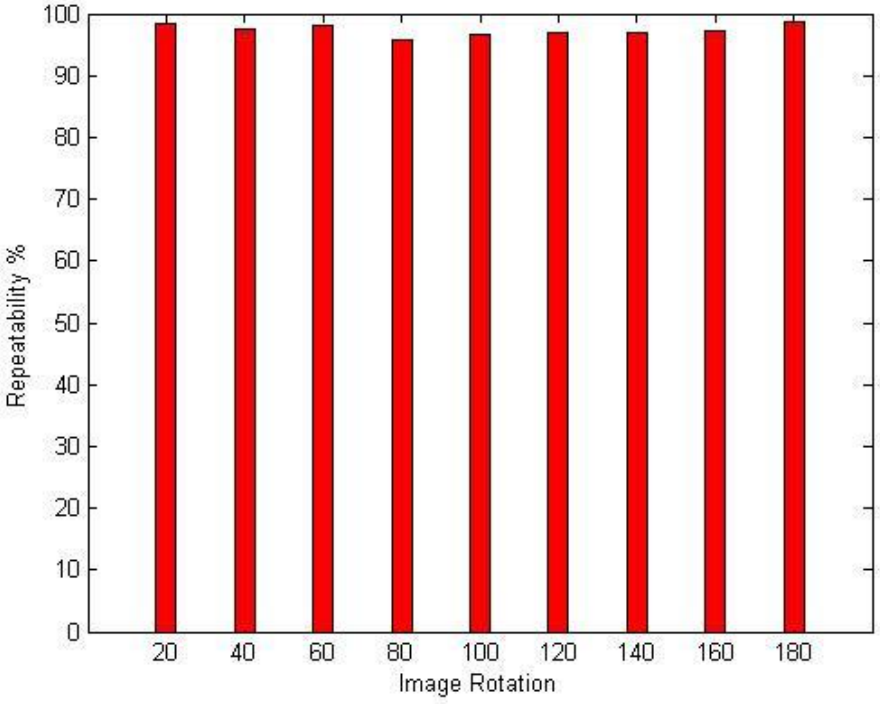


Figure 3.31 : Repeatability measure for modified SURF 36 and scale 0.2

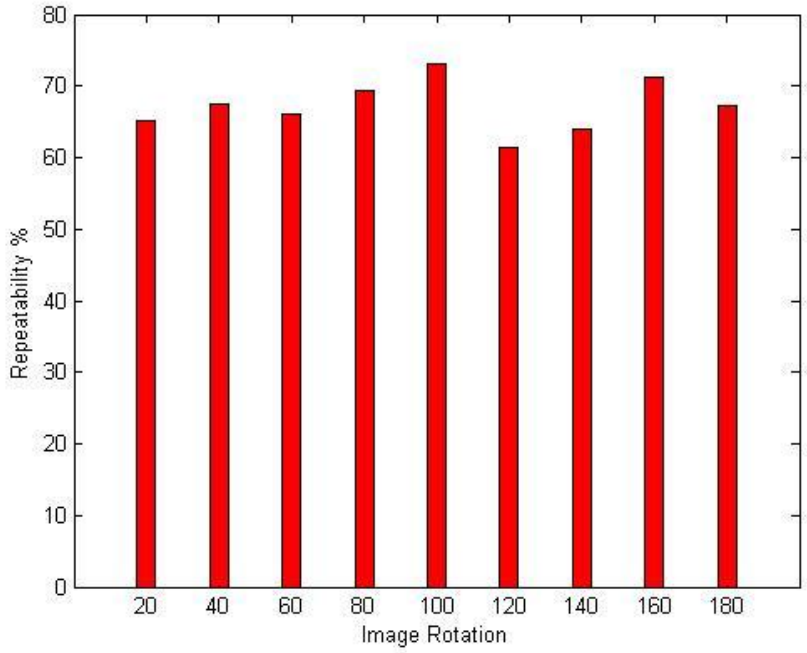


Figure 3.32 : Repeatability measure for descriptor length 64 and scale 0.4

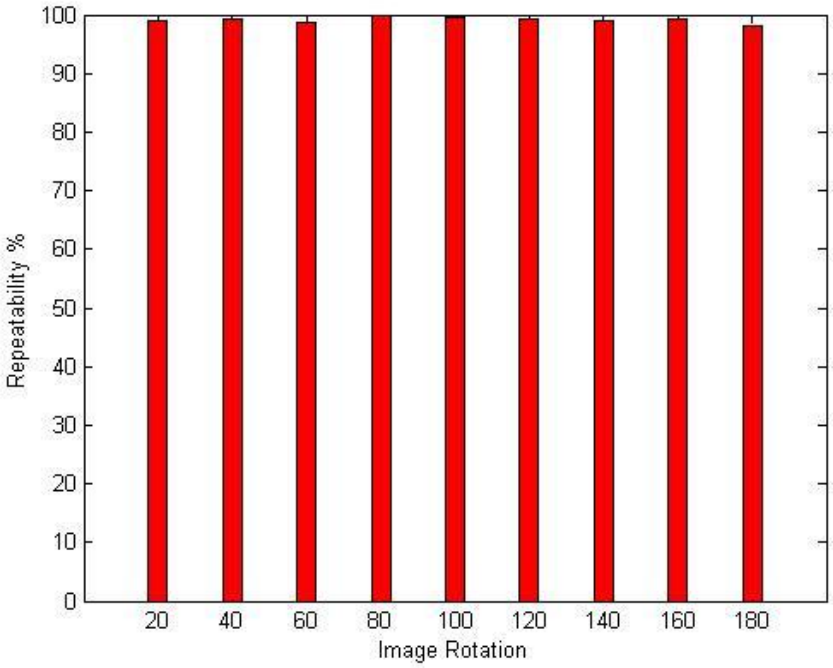


Figure 3.33 : Repeatability measure for modified SURF 36 and scale 0.4

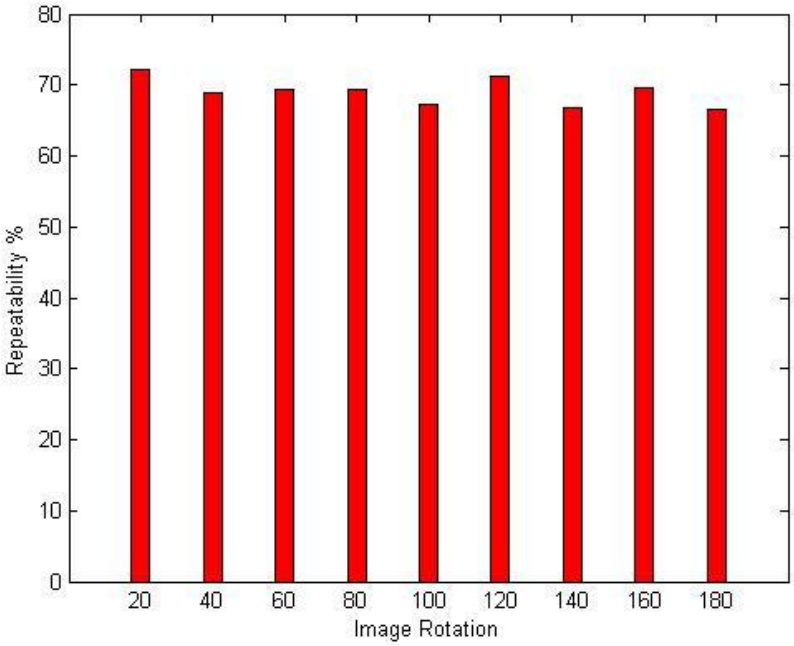


Figure 3.34 : Repeatability measure for descriptor length 64 and scale 0.6

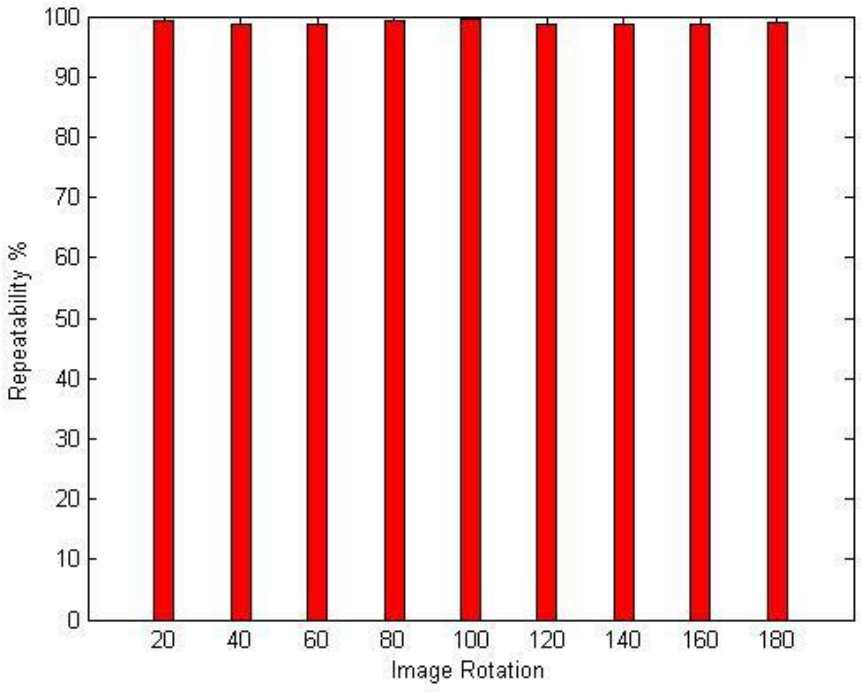


Figure 3.35 : Repeatability measure for modified SURF 36 and scale 0.6

More test results for the proposed modified SURF algorithm are presented in Appendix A for descriptor length 128 with scale variation from 0.2 to 1 and rotation angle 15°. Also, in Appendix A, more results for the developed modified SURF algorithm with descriptor lengths 36 ad 64 with scale variation from 0.2 to 1 and rotation angle 45° with 9×9 sample points I each sub-region.

Four GCPs are matched between the original image, as a representative of the database images, and a rotated and scaled image as representation of the real-time images captured from the UAV. As shown in the previous figures (Figure 3.20 to Figure 3.29), the proposed algorithm, with different descriptor length, has a robust performance against scale and rotation variation. The figures (Figure 3.30 to Figure 3.35) demonstrate that the performance of the interest point detection algorithm is improved when the descriptor length condition is reduced such that the number of sample points in each sub-region is increased (Sheta et al. 2012b).

Results clearly show that increasing number of samples in each subdivision is effective in the developed matching algorithm with high repeatability score up to 99%. The repeatability measure is used to provide a measure on the reliability of the developed algorithms for detecting the same interest points under different scale and rotation variations.

3.6 GPU CUDA implementation

In recent years, and because of the vast advances in the geomatics field, speeding up the processing time and solving computationally data-intensive problems were challenging. These challenges evolve, mainly, when we are dealing with some applications, that require real-time or near real-time processing. Organizing units of work into independent tasks and executing multiple tasks concurrently are traditional solutions for speeding up the process. For many years,

parallel processing was introduced as an efficient solution for this challenge using graphics processing unit (GPU). GPU is a specialized electronic circuit designed to rapidly manipulate and alter memory in such a way to accelerate the building of images in a frame buffer. However, traditional GPU development is based on graphics function library, for example OpenGL and Direct 3D, which makes the GPU used only by the professional people familiar with graphics API.

Recently, (Compute Unified Device Architecture (CUDA) was introduced as a new technology of general-purpose processing on GPU. CUDA mainly use C programming tools and compiler, which help to have better compatibility and portability making the development of general GPU programs easier.

In the coming sections, the image matching employed in the thesis will be implemented on GPU-CUDA platform . The processing time will be compared with the CPU implementation of the same matching algorithm when applied for multiple cores in the same computer, computer cluster and GPU.

3.6.1 General Purpose (GP) GPU principles

The implementation of programmable graphic processing units (GPU) as an alternative computation platform to the traditional use of the central processing unit (CPU) has been increased in areas where real-time application is needed. These GPUs performance is characterised with a peak performance of hundreds of GFLOPS which is an order of magnitude higher when compared to that of CPUs (Bjorke 2006). Streams of vertices and fragments (pixels) are processed independently in parallel. The advantage of data parallel SIMD (single instruction multiple data) architecture available in the GPU provides an abstraction for

performing general purpose computations on GPUs (GPGPU) and for treating the GPU as a stream processor.

In the GPGPU framework, the role of the computational kernels is performed via the fully programmable vertex and fragment processors while a memory model is employed for video memory (framebuffers, textures etc.). The CPU's random read-only memory interface equals texture mapping on the GPU while a memory-write mechanism is provided through the ability to render directly into texture (which is called off-screen rendering) .

Due to its specialized design, the GPU memory model is more restricted, where random memory writes are not allowed, when compared to a CPU. Texture memory caches design provides the necessary speed and prevention of concurrent read and write into the same memory address. Therefore, distinct read and write textures must be used allowing these operations to be exchanged after each render pass making the write texture available as input and vice versa.

For each algorithm implementation on the GPU, different computational steps are often transformed to different fragment programs. For each computational step, the appropriate fragment program is tied to the fragment processor and a render operation is invoked. A stream of fragments is generated by the rasterization engine which provides a fast way of interpolating numbers in graphics hardware. Multiple fragment programs are executed within multiple off-screen rendering passes by the GPGPU applications. A simple and efficient off-screen rendering mechanism in OpenGL is provided by Frame-Buffer Objects (FBOs), while pixel-buffers exist on older graphics cards.(Pharr and Fernando 2005) has more details about GPGPU programming.

3.6.2 Algorithm description

The interest point detection and descriptor building for the proposed algorithm can be summarised as follows:

1. Construct integral images through fast box filtering as introduced by (Viola and Jones 2001).
2. Create Hessian based scale space pyramid to find the interest point candidates.
3. Detect interest point and extract it from the candidate interest points using 3D non maximum suppression.
4. Determine a characteristic direction through orientation assignment to each interest point.
5. Build the descriptor for each interest point based on the characteristic direction (to ensure rotation invariance).
6. Normalize the description vector to achieve invariance to lighting conditions.

3.6.3 Algorithm implementation

3.6.3.1 Integral image computation

As mentioned in 3.2.1, to achieve efficient and fast computation, integral images approach is employed, for box-type convolution filters implementation. The integral image represents the summation of all pixel intensities in the input image I above and to the left of the location (x, y) as shown in Equation 3.1.

For the integral image computation on the GPU, it could be computed by two inclusive parallel-prefix sums: one along the rows, and one along the columns. Therefore, four kernel calls are used to implement the integral image on the GPU.

3.6.3.2 Interest point detection

Determinants calculation: this process is performed to each point in the image. This calculation is performed in parallel since the determinant calculation at one point is independent of the calculation at another point. One determinant calculation can be mapped to one GPU thread enabling more GPU threads to be available.

Interest Point Localization: where each point is defined from 8 adjacent pixels on the same scale and 9 pixels on the above and lower scales. This process is done by one GPU thread and therefore enabling more GPU threads to be available.

3.6.3.3 Interest point description

As for the orientation assignment process, for Haar response calculation, all these calculations are independent with each other and can be processed in parallel. The interest points are loaded into shared memory to minimize the memory bandwidth. Then, the Haar response is calculated to produce the rotation angle for each interest point $\theta = a\tan 2(d_y, d_x)$. The result is stored as (d_x, d_y, θ) and sorted using bitonic sort.

As for the dominant orientation calculation, an orientation window with size $\frac{\pi}{3}$ is used and all these windows are independent to each other and can be processed in parallel. After calculating the Haar wavelet response, the sum of magnitudes for d_x and d_y over the orientation window is calculated $\sqrt{(\sum d_x)^2 + (\sum d_y)^2}$ such that the largest vector magnitude has the dominant orientation with value $\varphi = -a\tan 2(\sum d_y, \sum d_x)$.

Descriptor Building: each interest point is described by the intensity distribution of its neighborhood and consequently these calculations are independent to each other and can be

processed in parallel. Two kernels are used for this step. The first one is assigned to the unnormalized descriptors and the other one to the normalized descriptors.

As for the unnormalized kernel, the number of blocks and threads vary depending on the descriptor length (36, 64, or 128) and the number of sample points in each sub-region as discussed in 3.3.2. For example, 16 blocks per interest point and 25 threads per block are allocated in the case of D_V with length 64 and number of sample points equals 25. In the case of D_V with length 36 and sample points in each sub-region 64, the number of blocks allocated is 9 and the number of threads is 64 per interest point.

As for the normalized kernel, one block and 64 threads are allocated for each interest point in the case of descriptor length equals 64 and one block and 36 threads in the case of descriptor length equals 36.

A summary for the implementation is shown in Figure 3.36.

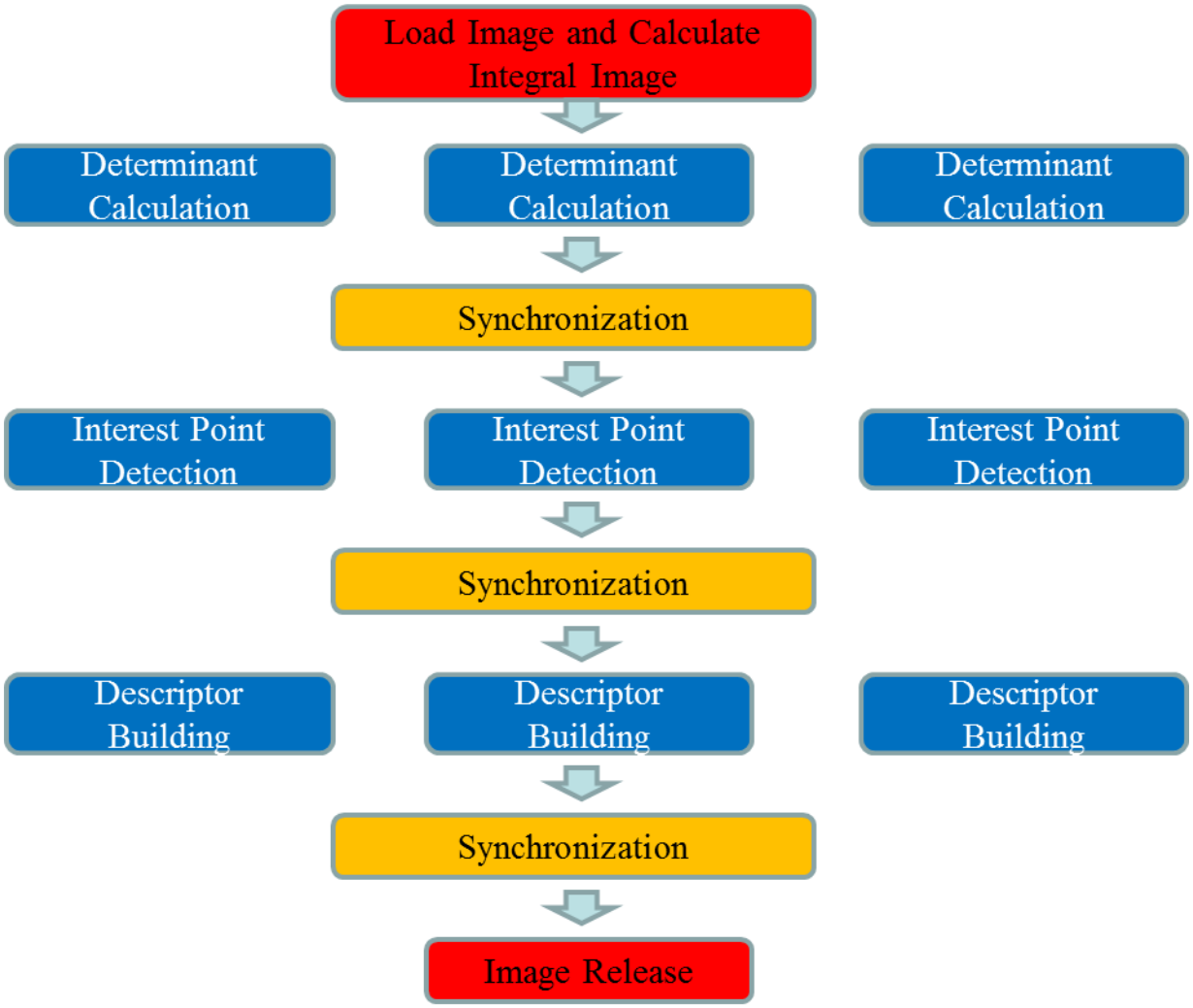


Figure 3.36: GPU CUDA SURF implementation (Sheta et al. 2012e)

3.7 Test set and results

The same data set used in section 3.5 is used with the following characteristics:

1. Images with sizes 640x480, 1024x768, and 1280x960.
2. GPU: Nvidia Geforce GT 540M, 97 cores.
3. CPU: Intel Core i7 Quad core 2760QM, 8G memory.

Tests were conducted using the proposed GPU CUDA approach and CPU based approach and the comparison is based on the computation time required to process all images of varying sizes with different approaches, percentage of time for computations feature point independent calculations using CUDA, and the overall speedup comparison between the CPU and CUDA implementations.

Summary of the timing performance for each component of the developed methodology is presented. Table 3-2 shows the number of interest point detected in the input image with size 1024 × 768 with the time taken for interest point independent processes computation. Results showed that the computational load is higher for interest point descriptor building because it includes the determination of the dominant orientation which is a high time consuming process.

Figure 3.39 shows the time required to process different images with sizes 640×480, 1024×768, and 1280×960. Results showed that speed up of the process time is linearly increased with the image size increase.

Table 3-4 shows the overall speedup comparison between different implementation of the developed methodology with different image sizes. Results clearly showed that GPU-CUDA implementation of the developed methodology speeded up the computation time with average 99.57%.

Figure 3.37: Input image with size 1024×768

Figure 3.38: Output image with interest points detected (Sheta et al. 2012e)

Table 3-2: Number of interest points detected using GPU CUDA with time taken for each process (Sheta et al. 2012e)

No. of Interest Points Detected	1293
Integral Image duration	6 msec
Interest Point Detection Duration	11 msec
Descriptor Building Duration	13 msec

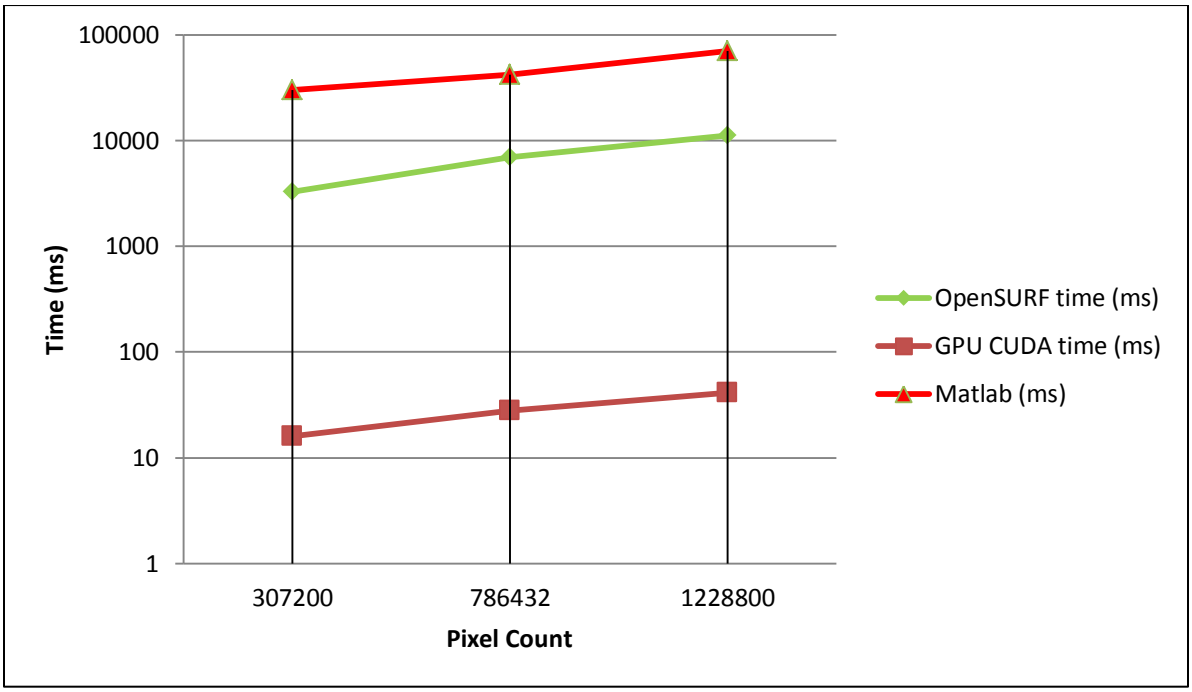


Figure 3.39: Computation time required to process all images of varying sizes with different approaches (Sheta et al. 2012e)

Table 3-3: Percentage of time for computations feature point independent calculations using CUDA (Sheta et al. 2012e)

Image Size	640×480	1024×768	1280×960
Integral Image	29.41%	20%	32.14%
Interest Point Detection	29.41%	36.66%	33.92%
Descriptor Building	41.18%	43.34%	33.94%

Table 3-4: Overall speedup comparison between the CPU and CUDA implementations (Sheta et al. 2012e)

Image Size	640×480	1024×768	1280×960
Feature Count	564	1293	1893
GPU CUDA	16 msec	28 msec	41 msec
CPU implementation	3257 msec	6926 msec	11237 msec
CPU implementation Speed Up	99.5 %	99.59 %	99.63%

3.8 Summary

In this chapter, we have introduced and tested the matching algorithm with descriptor length 36 as the matching algorithm for VBN depending on a lower number of interest point matches between real-time captured images and those from a database. Additionally, the samples count in the sub-divisions with the different descriptor length (36, 64, and 128) was changed to test the effect of the number of samples in each subdivision on the accuracy of the matching algorithm. Results showed that a number of samples are effective in the matching algorithm, which had previously not been investigated.

The proposed algorithm in this chapter has been implemented on CUDA platform which showed a fast implementation employing on the shelf graphics card (Sheta et al. 2012e). Implementation using CUDA satisfies the real-time processing requirements, while keeping the high accuracy which is suitable for real-time application such as VBN. The GPU can be manipulated into efficient image processing that is faster than the CPU, and also frees the CPU for other activities. Low parallelism computations can sometimes be faster than transferring back and forth to host. GPU can achieve superior results compared to CPU processing times providing that:

1. Maximizing the independent parallelism in the implemented algorithm.
2. Applying more computation on the GPU to avoid costly data transfers time loss.
3. Partitioning the computation to keep the GPU multiprocessors equally busy.
4. Taking advantage of shared memory.

Chapter Four: **NONLINEAR OPTIMIZATION APPROACHES FOR OBJECT SPACE TRANSFORMATION PARAMETERS ESTIMATION**

The main objective of this chapter is to investigate the estimation of the georeferencing parameters necessary for VBN of aerial vehicles with the assumption that the UAV experience large values of the rotational angles, which will lead to nonlinearity of the estimation model. In this case, traditional least squares approaches will fail or will take long time to estimate the georeferencing parameters, because of the expected nonlinearity of the mathematical model. Five different nonlinear least squares methods are presented for estimating the transformation parameters – this include four gradient based nonlinear least squares methods (Trust region, Trust region dogleg algorithm, Levenberg-Marquardt, and Quasi-Newton line search method) and one non-gradient method (Nelder-Mead simplex direct search) is employed for the six transformation parameters estimation process.

4.1 Introduction

Although a robust scale and rotation invariant image matching algorithm is important for VBN of aerial vehicles, the proper estimation of the object space transformation parameters improves the efficiency of the navigation process by the real-time estimation of transformation parameters. These parameters can then be used in aiding the inertial measurements data in the navigation estimation filter.

A pose estimation problem can be investigated through the correspondence between the 3D Ground Control Points (GCP) and the corresponding 2D points in image coordinates (Lu et al. 2000). Gauss-Newton and Levenberg-Marquardt, among other methods such as Trust Region method, are considered as classical iterative approaches for solving the nonlinear least square problem for pose estimation problem (Lowe 1991).

General unconstrained nonlinear optimization problem can be described by the following equation (Teunissen 1990):

$$\min F(x), x \in R^n, F: R^n \to R \qquad \textbf{4.1}$$

where

x unknown parameters to be estimated (object space parameters)

$F(x)$ objective function

R^n n unknowns belong to the Real number

Iterative algorithms are implemented to solve for minimization of this problem. The main goal is to find a new value to the unknown parameters (vector x) that lead to minimizing the objective function value (Venkataraman 2002).

General solution of the minimization problem can be reached through the following scheme:

$$x_{k+1} = x_k + t_k d_k, k = 0,1,2,... \qquad \textbf{4.2}$$

where

x_{k+1}, x_k Estimated unknown parameters at different iteration steps k+1, k respectively

t_k Positive scalar

d_k Search direction vector

For the purpose of the current study of the different optimization algorithms, the following steps can be followed (Teunissen 1990):

1. Start the iteration at k=0 with initial values for the unknown parameters.
2. Determine the search direction vector d_k.
3. According to the line search strategy, determine t_k such that $F(x_{k+1}) \leq F(x_k)$

4. Threshold is predefined such that the obtained solution must satisfy it.
5. Stop the iteration when threshold condition is satisfied and x_{k+1} will be the solution for equation 4.1. Otherwise, *k* is increased by one and repeat the procedure from steps 2 to 5.

Depending on this general procedure, the main difference between the optimization methods used is the choice of the search direction vector d_k and the scalar t_k.

When the objective function is the weighted sum of squares such that $F(x) = \frac{1}{2}\|y - H(x)\|^2$, the least squares approach is used (El-Habiby et al. 2009) and the traditional Newton's method provides a solution for the search direction d_k such that:

$$\begin{aligned} J(x_k)d_k &= -F(x_k) \\ x_{k+1} &= x_k + d_k \end{aligned} \qquad \textbf{4.3}$$

where:

$J(x_k)$ is the *n* by *n* Jacobian matrix

$$J(x_k) = \begin{bmatrix} \nabla F_1(x_k)^T \\ \nabla F_2(x_k)^T \\ \dots \\ \dots \\ \nabla F_n(x_k)^T \end{bmatrix}$$

However, this method has some difficulties especially when $J(x_k)$ is singular. In this case, the Newton step d_k is undefined. Additionally, Newton's method may not converge if the starting point is far from the solution.

VBN is based on localizing set of features (with known coordinates) on the ground and finding their matches in the image taken by an imaging sensor on the UAV. Then, through Collinearity

equations, object space transformation parameters are estimated such that these matches are transformed into position information.

Given the above, two challenges in VBN are addressed in this thesis. The first challenge is when large tilted aerial platform is used which leads to oblique images with large tilt angles leading to nonlinearity of the estimation model. In this case, traditional least squares approaches will fail or take very long time to estimate the object space transformation parameters, because of the expected nonlinearity of the mathematical model. The limitations in traditional least square approach for solving the collinearity equations to deal with this situation have been presented and the solution to this problem is introduced using five different nonlinear least squares methods. These methods are Trust region, Trust region dogleg algorithm, Levenberg-Marquardt, Nelder-Mead simplex direct search, and Quasi-Newton line search method.

The second challenge is the number of matches necessary for solving the collinearity equations which is highly required to be minimal as possible meet the real-time requirements for the proposed UAV VBN approach.

4.2 Mathematical model for Collinearity equations

In a VBN system, 3D position and attitude of the aerial vehicle can be estimated by means of the conjugate features in the UAV aerial image and geo-referenced satellite image. Collinearity equations are used for mathematical transformation between 2D Image and 3D object space stored in the database. Assuming perspective geometry and neglecting the distortion parameters, this relation is expressed as shown in Figure 4.1and equation 4.4:

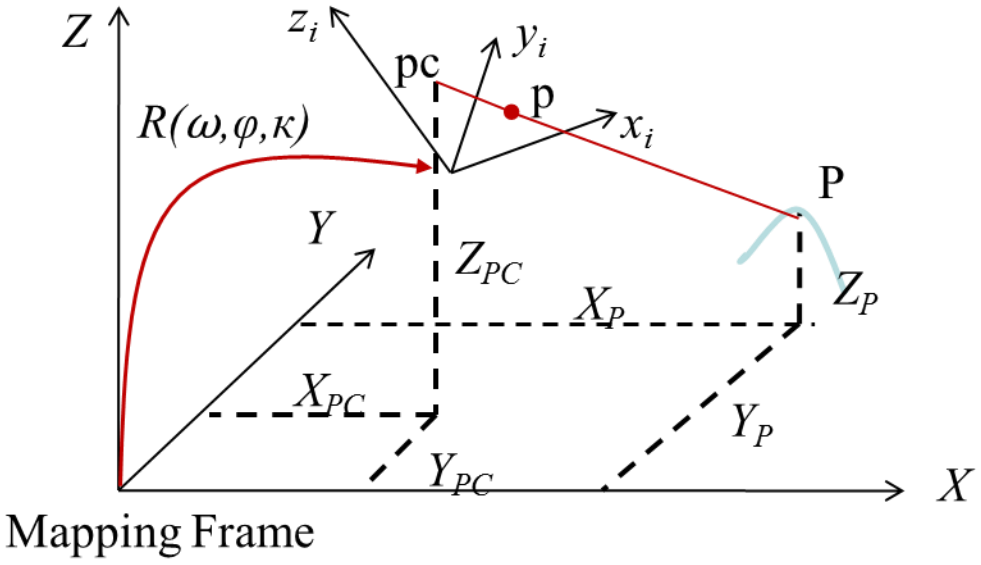


Figure 4.1: Collinearity equations model

$$x_p = x_{pc} - f\frac{c_{11}(X_P - X_{PC}) + c_{21}(Y_P - Y_{PC}) + c_{31}(Z_P - Z_{PC})}{c_{13}(X_P - X_{PC}) + c_{23}(Y_P - Y_{PC}) + c_{33}(Z_P - Z_{PC})}$$

$$y_p = y_{pc} - f\frac{c_{12}(X_P - X_{PC}) + c_{22}(Y_P - Y_{PC}) + c_{32}(Z_P - Z_{PC})}{c_{13}(X_P - X_{PC}) + c_{23}(Y_P - Y_{PC}) + c_{33}(Z_P - Z_{PC})} \qquad \textbf{4.4}$$

Where:

(x_p, y_p) are the image coordinates.

(X_P, Y_P, Z_P) are the object database ground coordinates of the conjugate points (can be determined after the matching procedure)

(x_{pc}, y_{pc}, f) are the camera coordinates of the principal point and principal distance which are known as the Interior Orientation Parameters (IOP)s (can be determined by calibration of the camera before the mission) .

(X_{PC}, Y_{PC}, Z_{PC}) are the unknown parameters which define the position of projection centre of the camera body in ground coordinate system.

$c_{11}, ..., c_{33}$ are the elements of the 3D rotation matrix R containing the rotation angles ω, φ, and κ between the camera axes and the mapping coordinate system, such that:

$$R = \begin{pmatrix} \cos\varphi\cos\kappa & -\cos\varphi\sin\kappa & \sin\varphi \\ \cos\omega\sin\kappa + \sin\omega\sin\varphi\cos\kappa & \cos\omega\cos\kappa - \sin\omega\sin\varphi\sin\kappa & -\sin\omega\cos\varphi \\ \sin\omega\sin\kappa - \cos\omega\sin\varphi\cos\kappa & \sin\omega\cos\kappa + \cos\omega\sin\varphi\sin\kappa & \cos\omega\cos\varphi \end{pmatrix}$$

Using Taylor's theorem to linearize equation 4.4, these equations can be rewritten as:

$$F = x_{pc} - f\frac{m}{q} = x_p$$
$$G = y_{pc} - f\frac{s}{q} = y_p \qquad \textbf{4.5}$$

where:

$$q = c_{31}(X_P - X_{PC}) + c_{32}(Y_P - Y_{PC}) + c_{33}(Z_P - Z_{PC})$$
$$m = c_{11}(X_P - X_{PC}) + c_{12}(Y_P - Y_{PC}) + c_{13}(Z_P - Z_{PC})$$
$$s = c_{21}(X_P - X_{PC}) + c_{22}(Y_P - Y_{PC}) + c_{23}(Z_P - Z_{PC})$$

Applying Taylor's theorem to equation 4.5, taking partial derivatives with respect to the unknowns, those equations are expressed as:

$$F_0 + (\frac{\partial F}{\partial \omega})_0 d\omega + (\frac{\partial F}{\partial \varphi})_0 d\varphi + (\frac{\partial F}{\partial \kappa})_0 d\kappa + (\frac{\partial F}{\partial X_{PC}})_0 dX_{PC} + (\frac{\partial F}{\partial Y_{PC}})_0 dY_{PC} + (\frac{\partial F}{\partial Z_{PC}})_0 dZ_{PC} +$$
$$(\frac{\partial F}{\partial X_P})_0 dX_P + (\frac{\partial F}{\partial Y_P})_0 dY_P + (\frac{\partial F}{\partial Z_P})_0 dZ_P = x_p \qquad \textbf{4.6}$$

$$G_0+(\frac{\partial G}{\partial \omega})_0 d\omega+(\frac{\partial G}{\partial \varphi})_0 d\varphi+(\frac{\partial G}{\partial \kappa})_0 d\kappa+(\frac{\partial G}{\partial X_{PC}})_0 dX_{PC}+(\frac{\partial G}{\partial Y_{PC}})_0 dY_{PC}+(\frac{\partial G}{\partial Z_{PC}})_0 dZ_{PC}+$$
$$(\frac{\partial G}{\partial X_P})_0 dX_P+(\frac{\partial G}{\partial Y_P})_0 dY_P+(\frac{\partial G}{\partial Z_P})_0 dZ_P=y_p \qquad \textbf{4.7}$$

where:

F_0, G_0 are calculated at the initial values for the unknown parameters in the collinearity equations. $(\frac{\partial F}{\partial \omega})_0, ..., (\frac{\partial G}{\partial \varphi})_0$ are calculated at the initial values for the unknown parameters in the collinearity equations.

$d\omega, d\varphi, ..., dZ_P$ are the corrections to be added to the initial values for the unknown parameters.

If the least square approach is used, and noticing that x_p and y_p are measured values, residual terms should be added to the equations. Therefore, the linearized equations after adding these residuals can be expressed as:

$$b_{11}d\omega+b_{12}d\varphi+b_{13}d\kappa-b_{14}dX_{PC}-b_{15}dY_{PC}-b_{16}dZ_{PC}+b_{14}dX_P+b_{15}dY_P+b_{16}dZ_P=J+v_{xp} \qquad \textbf{4.8}$$

$$b_{21}d\omega+b_{22}d\varphi+b_{23}d\kappa-b_{24}dX_{PC}-b_{25}dY_{PC}-b_{26}dZ_{PC}+b_{24}dX_P+b_{25}dY_P+b_{26}dZ_P=K+v_{yp} \qquad \textbf{4.9}$$

where:

$b_{11}, ..., b_{26}, J \text{ and } K$ are given as:

$$b_{11}=\frac{f}{q^2}[m(-c_{33}\Delta Y+c_{32}\Delta Z)-q(-c_{13}\Delta Y+c_{12}\Delta Z)]$$

$$b_{12} = \frac{f}{q^2}[m(\cos\varphi\Delta X + \sin\omega\sin\varphi\Delta Y - \cos\omega\sin\varphi\Delta Z) - q(-\sin\varphi\cos\kappa\Delta X + \sin\omega\cos\varphi\cos\kappa\Delta Y - \cos\omega\cos\varphi\cos\kappa\Delta Z)]$$

$$b_{13} = \frac{-f}{q}(c_{21}\Delta X + c_{22}\Delta Y + c_{23}\Delta Z)$$

$$b_{14} = \frac{f}{q^2}(mc_{31} - qc_{11})$$

$$b_{15} = \frac{f}{q^2}(mc_{32} - qc_{12})$$

$$b_{16} = \frac{f}{q^2}(mc_{33} - qc_{13})$$

$$J = x_p - x_{pc} + f\frac{m}{q}$$

$$b_{22} = \frac{f}{q^2}[s(\cos\varphi\Delta X + \sin\omega\sin\varphi\Delta Y - \cos\omega\sin\varphi\Delta Z) - q(\sin\varphi\cos\kappa\Delta X - \sin\omega\cos\varphi\sin\kappa\Delta Y + \cos\omega\cos\varphi\sin\kappa\Delta Z)]$$

$$b_{23} = \frac{f}{q}(c_{11}\Delta X + c_{12}\Delta Y + c_{13}\Delta Z)$$

$$b_{24} = \frac{f}{q^2}(sc_{31} - qc_{21})$$

$$b_{25} = \frac{f}{q^2}(sc_{32} - qc_{22})$$

$$b_{26} = \frac{f}{q^2}(sc_{33} - qc_{23})$$

$$K = x_p - x_{pc} + f\frac{s}{q}$$

$$\Delta X = X_P - X_{PC}$$

$$\Delta Y = Y_P - Y_{PC}$$

$$\Delta Z = Z_P - Z_{PC}$$

From the previous equations, the number of necessary points (n) for the object space transformation parameters in the collinearity equations should satisfy $n \geq 3$. If $n \geq 3$ is realized, the number of linearized equations to be solved is 2n with 6 unknowns to be estimated. These equations in the matrix form can be expressed as $A\Delta x = L + V$, where

$\Delta x = \begin{bmatrix} d\omega & d\varphi & d\kappa & dX_{PC} & dY_{PC} & dZ_{PC} \end{bmatrix}^T$ can be calculated using least square approach from $(A^T A)\Delta x = (A^T L)$.

The determination of parameters through minimization can be considered as the definition of a general optimization problem. Nonlinear least squares optimization algorithms can be investigated through two general approaches. The first one through linearization of the model, where corrections to the unknown estimated parameters are added during the iteration steps on the assumption of local linearity, as shown above. Linearization of the Collinearity equations for space parameters estimation requires repeated computations to improve the initial values assigned to the transformation parameters and this approach is clearly investigated in (Wolf and Dewitt 2000). However, nonlinear least squares optimization approaches are used in the general case where the aerial platform will lead to non-vertical images. The second one is based on modifications made to the steepest descent approach, which is a gradient method. However, for optimal solution, an optimal interpolation between the two methods can be used for the representation of the nonlinear model (Marquardt 1963).

4.3 Nonlinear optimization algorithms

As mentioned before, five optimization methods will be used to estimate the object space parameters estimation (*X*, *Y*, and *Z* translation vector and rotation angles ω, φ, and κ).

As mentioned in 4.2, the main difference between the optimization methods used is the choice of the search direction vector d_k and the scalar t_k (El-Habiby et al. 2009) as well as the objective function itself or its derivative.

4.3.1 Trust region algorithm

The trust region approach for nonlinear optimization has been studied over 50 years (Conn et al. 2000). This method is basically considered as one of the basic descent methods.

At each iteration point x_k, define a model $m_k(x)$ to approximate the objective function within a suitable neighbourhood of x_k. This neighbourhood is the trust region. The set of all points defining the trust region is found in the following equation:

$$\beta_k = \left\{ x \in R^n \quad , \| x - x_k \|_k \leq \Delta_k \right\} \qquad \textbf{4.10}$$

where:

Δ_k is the trust region radius.

$\|.\|_k$ is iteration dependent norm.

The basic trust region algorithm can be summarized as follows (Conn et al. 2000) :

1. Start with the initial point x_0 and initial trust region radius Δ_0.
2. Define a set of constants $\eta_1, \eta_2, \gamma_1, and \gamma_2$ satisfying this condition $0 < \eta_1 \leq \eta_2 < 1$ and $0 < \gamma_1 \leq \gamma_2 < 1$.
3. Calculate $f(x_0)$ and set $k = 0$.
4. Choose $\|.\|_k$ and define the approximation model m_k in β_k.
5. Calculate a step s_k to minimize the model m_k satisfying $x_k + s_k \in \beta_k$.

6. Calculate $f(x_k + s_k)$ and define a parameter (ρ) which is $\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$. If $\rho_k \geq \eta_1$, then set $x_{k+1} = x_k + s_k$; otherwise, set $x_{k+1} = x_k$.

7. Update the trust region radius according to $\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2 \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2) \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1 \end{cases}$.

8. Increment *k* by 1 and repeat the process.

The trust region method is considered a gradient based method where the search direction will be evaluated using the gradient of the objective function. In this method, the model $m_k(x)$ is quadratic model of the objective (based on the second-order Taylor series of *f* at x_k).This helps to minimize the number of iterations required for the convergence (Sheta et al. 2012d).

4.3.2 Trust region dogleg algorithm

At each iteration point x_k, the trust region sub-problem where the quadratic model is used can be expressed as:

$$m_k(x_k + s_k) = f(x_k) + \langle g_k, s_k \rangle + \frac{1}{2} \langle s_k, \mathcal{H}_k s_k \rangle \qquad \textbf{4.11}$$

where:

g_k is the gradient $\nabla f(x_k)$.

$\mathcal{H}_k$ is the Hessian approximation matrix.

This algorithm is based on computing the step *d*, which minimizes the model described above using the Powell dogleg procedure. For a detailed description about Powell method, see (Powell and Atomic Energy Research Establishment 1968).

The step *d* is found from a convex combination of a Cauchy step (a step along the steepest descent direction) and a Gauss-Newton step for *f(x)*. The Cauchy step is calculated as:

$$d_c = -\alpha J(x_k)^T F(x_k) \qquad \textbf{4.12}$$

Where α is chosen to minimize the model described.

The Gauss-Newton step is found by:

$$J(x_k) d_{GN} = -F(x_k) \qquad \textbf{4.13}$$

The step *d* is chosen such that:

$$d = d_c + \mu(d_{GN} + d_c) \qquad \textbf{4.14}$$

where μ is the largest value in the interval [0,1] such that $\|d\| \leq \Delta$.

The trust region dogleg method is a gradient based optimization method. The cost of computing the dogleg paths in addition to the Cauchy and the Gauss-Newton steps increases the number of iterations more than the trust region approach.

4.3.3 Levenberg-Marquardt algorithm

In this algorithm, a scalar α is introduced to help the choice of the magnitude and the direction of the descent such that the Gauss-Newton algorithm is modified using the trust region approach. Determination of using either the Gauss-Newton method direction or the steepest descent method direction is done by this scalar. The search direction is computed by (Luenberger 1984):

$$\left[J^T C_y^{-1} J + \alpha I \right] d_k = -JF(x_k) \qquad \textbf{4.15}$$

where:

C_y is positive definite observation covariance matrix

When $\alpha = 0$, the algorithm will be the Gauss-Newton approach and when α tends to have large values, it will require the steepest descent method. This factor α makes the algorithm moves smoothly between Gauss-Newton and steepest descent method. When the current estimate of the variables leads to minimization of the objective function, the algorithm continuously moves to the Gauss-Newton approach. This leads to a decrease in the number of iterations needed for convergence.

Calculating the size of α, such that the algorithm is efficient in each iteration step, is the challenge in the Levenberg-Marquardt algorithm. The relative nonlinearity of the objective function $F(x)$ is estimated by a linear predicted sum of squares $f_{lps}(x_k)$ to control the step size (Sheta et al. 2012c):

$$f_{lps}(x_k) = J(x_{k-1})d_{k-1} + F(x_{k-1}) \qquad \textbf{4.16}$$

An estimate of the minimum $f_{ci}(x)$ is cubically interpolated, where $f_{ci}(x)$ is calculated by cubically interpolating $f(x_k)$ and $f(x_{k-1})$ (Marquardt 1963).

When $f_{lps}(x_k)$ is greater than $f_{ci}(x)$, reduce α or, to ensure that $f(x_{k+1}) < f(x_k)$ at each iteration step, it is increased such that the descent algorithm towards the solution is maintained (Nocedal and Wright 1999; Fletcher 2000).

4.3.4 Quasi-Newton line search algorithm

The basic idea behind this method is that approximation of the Hessian matrix $\mathcal{H}$ of the objective function is achieved by a positive definite matrix. This positive definite matrix is initially computed by $\mathcal{H}_0$. As the search continues, the approximating matrix is updated at each iteration step and consequently the second derivative information is updated and improved. The

Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update is calculated through the following (Broyden 1970):

1. Start with an initial point x_i and an approximation Hessian matrix $\mathcal{H}_k$.
2. Calculate the search direction d_k by solving: $\mathcal{H}_k d_k = -\nabla f(x_k)$.
3. Find an acceptable step size α_k in the search direction found in the previous step through a line search, then update $x_{k+1} = x_k + \alpha_k d_k$.
4. Let $s_k = \alpha_k d_k$.
5. Let $v_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.
6. Calculate $\mathcal{H}_{k+1} = \mathcal{H}_k + \frac{v_k v_k^T}{v_k^T s_k} - \frac{\mathcal{H}_k s_k (\mathcal{H}_k s_k)^T}{s_k^T \mathcal{H}_k s_k}$

Practically, $\mathcal{H}_0$ can be initialized with $\mathcal{H}_0 = I$, so that the first step will be equivalent to a gradient descent, but further iterations are refined by $\mathcal{H}_k$.

In this method, instead of computing the true Hessian as in Newton's method, an approximation that is based on the change in gradient between iterations will be used. The primary advantage is that there is no need to compute the exact Hessian at each point, which may be computationally expensive (Sheta et al. 2012d).

4.3.5 Nelder-Mead simplex direct search algorithm

This algorithm was first published in 1965 and is used in multidimensional unconstrained minimization problems. Since this algorithm achieves minimization for the objective function based on the function values without any derivative information, it is categorized in the direct search methods (Lagarias et al. 1998).

A simplex in n-dimensional space is created and represented by n+ 1 distinct vector, where n is the length of unknown x to be evaluated to minimize the objective function.

From the original paper of Nelder-Mead (Nelder and Mead 1965), the objective function *f(x)* is minimized through using four coefficients, where they should satisfy particular conditions. These parameters are reflection (ρ_r), expansion (χ), contraction (γ_c), and shrinkage (ξ).

This algorithm can be described as follows:

1. **Order:** based on the vertices values at: $f(x_1) \le f(x_2) \cdots \le f(x_{n+1})$.
2. **Reflection:** the reflection point is evaluated $x_r = \tilde{x} + \rho_r(\tilde{x} - x_{n+1})$ where $\tilde{x} = \sum_{i=1}^{n} \frac{x_i}{n}$ is the centroid of all vertices except x_{n+1}. Evaluate $f(x_r)$. If $f(x_1) \le f(x_r) < f(x_n)$, the reflected point x_r is accepted and the iteration is terminated.
3. **Expansion:**
    a. If $f(x_r) < f(x_1)$, the expanded point x_e is evaluated.
    b. $x_e = \tilde{x} + \chi(x_r - \tilde{x}) = \rho_r \chi(\tilde{x} - x_{n+1})$. Evaluate $f(x_e)$. If $f(x_e) < f(x_r)$, replace the worst point x_{n+1} with the expanded point x_e, and stop the iteration.
    c. If $f(x_e) \ge f(x_r)$, change the worst point x_{n+1} with the reflected point x_r, and stop the iteration.
4. **Contraction:**

a. If $f(x_n) \le f(x_r) < f(x_{n+1})$, evaluate the contacted point $x_c = x + \gamma_c(x_r - \tilde{x})$. If $f(x_c) \le f(x_r)$, change the worst point x_{n+1} with the contracted point x_c, and stop the iteration. Else go to step 5.

b. If $f(x_r) \ge f(x_{n+1})$, evaluate the contacted point $x_c = \tilde{x} - \gamma_c(\tilde{x} - x_{n+1})$.

c. If $f(x_c) \le f(x_{n+1})$, change the worst point x_{n+1} with the contracted point x_c and stop the iteration. Else go to step 5.

5. **Reduction:** evaluate f at n points $x_i = x_1 + \xi(x_i - x_1)$ for all i=2,…,n+1.

Since this method is basically a non-gradient optimization method, it is predicted that this method will fail due to its dependency on the objective function only without any derivative information.

4.4 Test set and results

Tests were conducted on the optimization algorithms using simulated data starting with the near-vertical aerial photo assumption captured with a 152.916 mm focal length camera. The tests were done using three ground control points 1 through 3 listed in Table 4-2. Image coordinates and ground control coordinates of the three points are listed in Table 4-2.

The real-time nonlinearity problem of the collinearity equations for proper estimation of the object space transformation parameters is investigated using the developed nonlinear optimization approaches.

Five nonlinear optimization approaches were introduces for estimating object space transformation parameters, which are necessary for the process of pose estimation of UAV VBN approach. Convergence of these methods were introduced using the near vertical assumption and

with the general tilted photograph case. The developed nonlinear optimization approaches are tested with general tilted photograph with different values for angle φ varying from $\varphi = 0°$ to 45° with step 5°. These methods are compared with respect to number of iterations and accuracy. The methods introduced are basically categorized as gradient methods (Trust Region, Trust Region Dogleg, Levenberg-Marquadrt method, and Quasi-Newton Line Search method) and non-gradient methods (Nelder-Mead simplex direct search method).

Results show that Nelder-Mead method has failed due the dependency on the objective function only without any derivative information.

As for gradient methods, tests showed different results in solving for the object space parameters based on the initial value for φ. As for trust region dogleg method, good results were achieved till $\varphi = 20°$ and then the number of iterations increased and the algorithm didn't converge since the degree of the nonlinearity was increased and the computed step d failed to minimize the model described in equation 4.11. As for the Levenberg-Marquardt approach, it is as a modified Gauss-Newton algorithm using the trust region approach where a scalar is used to help the choice of the magnitude and the direction of the descent. Depending on this scalar, either the Gauss-Newton method direction or the steepest descent method direction is determined. Therefore, the Levenberg-Marquardt method converged when $\varphi = 0°$. However, it didn't converge once φ is increased from 0°. This was predicted since the relative nonlinearity of the objective function $F(x)$, shown in equation 4.15, is estimated by a linear predicted sum of squares $f_{lps}(x_k)$ to control the step size which will fail to converge as φ increases above 0°. As for the Quasi-Newton method, it converged for all values of φ since this algorithm is based on

the Hessian matrix of the Collinearity equations which is updated using the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update .This update is function of the gradient.

As for the number of GCP needed for convergence, the object space parameters were successfully estimated using three GCP, which will assess in the image matching approach for solving for the necessary points needed for convergence of the collinearity equations parameters estimation problem (Sheta et al. 2012a).

Table 4-1: Initial Values for the Object Space Parameters

	Omega (rad)	Phi (rad)	Kappa (rad)	X_L (m)	Y_L (m)	Z_L (m)
Value	0	0	1.7947	1009.923	1038.056	649.614

Table 4-2: Image Points and Ground Control Points (GCP) Coordinates

	Image coordinates (mm)		Ground Control Coordinates (m)		
Point	x	y	X	Y	Z
1	86.421	-83.977	1268.102	1455.027	22.606
2	-100.916	92.582	732.181	545.344	22.299
3	-98.322	-89.161	1454.553	731.666	22.649

Table 4-3: Performance of optimization methods with initial φ (deg) = 0

Object Space parameters	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line	Nelder-Mead simplex direct

errors				search	search
$\delta\omega$ (rad)	0.059	0.076	0.008	0.051	0.007
$\delta\varphi$ (rad)	0.021	0.003	0.021	0.050	0.021
δK (rad)	0.004	0.011	0.0004	0.048	0.017
δX_L (m)	21.50	17.93	17.933	17.933	19.669
δY_L (m)	6.274	6.058	6.057	6.058	57.502
δZ_L (m)	0.361	1.416	1.417	1.416	4.502
No. of Iteration	4	2	1	2	158

Table 4-4: Performance of optimization methods with initial φ (deg) = 5

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.065	0.059	0.029	0.049	0.007
$\delta\varphi$ (rad)	0.026	0.058	0.731	0.050	0.072
δK (rad)	0.004	0.026	0.647	0.045	0.002
δX_L (m)	21.046	44.462	297.345	17.933	127.438
δY_L (m)	6.384	9.413	365.369	6.058	17.553
δZ_L (m)	0.13	12.096	535.928	1.416	63.175
No. of Iteration	2	2	55	2	142

Table 4-5: Performance of optimization methods with initial φ (deg) = 10

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.058	0.047	0.032	0.045	0.007
$\delta\varphi$ (rad)	0.030	0.109	0.728	0.050	0.148
δK (rad)	0.008	0.063	0.644	0.037	0.018
δX_L (m)	23.233	78.689	297.885	17.933	166.201
δY_L (m)	6.672	14.67	362.236	6.058	11.388
δZ_L (m)	1.262	31.887	532.772	1.417	33.672
No. of Iteration	3	3	48	2	153

Table 4-6: Performance of optimization methods with initial φ (deg) = 15

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.055	0.063	0.033	0.041	0.007
$\delta\varphi$ (rad)	0.036	0.097	0.725	0.050	0.235
δK (rad)	0.021	0.068	0.641	0.025	0.006
δX_L (m)	31.988	78.031	298.405	17.933	228.742
δY_L (m)	7.493	18.594	359.337	6.058	12.639

δZ_L (m)	5.905	31.444	529.787	1.417	37.796
No. of Iteration	4	3	44	3	142

Table 4-7: Performance of optimization methods with initial φ (deg) = 20

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.053	0.049	0.035	0.032	0.007
$\delta\varphi$ (rad)	0.090	0.198	0.723	0.049	0.305
δK (rad)	0.053	0.137	0.638	0.005	0.020
δX_L (m)	66.836	142.211	298.797	17.933	280.881
δY_L (m)	11.709	36.664	357.216	6.058	18.823
δZ_L (m)	26.114	74.913	527.565	1.417	62.472
No. of Iteration	6	4	42	3	155

More test results for the developed nonlinear optimization approaches are presented in Appendix B. The developed nonlinear optimization approaches are tested with general tilted photograph with different values for angle φ varying from φ = 25° to 45° with step 5°.

4.5 Summary

Real-time nonlinearity problem of the collinearity equations for proper estimation of the object space transformation parameters (X, Y, and Z translation vector and rotation angles ω, φ, and κ) has been investigated. Tests were conducted with general tilted photograph with different values for only angle φ varying from $\varphi = 0°$ to 45° with step 5° as an example for high rotation angle which will lead to nonlinearity of the estimation model. The developed methods covered both gradient and non-gradient based nonlinear optimization approaches. As for the gradient based nonlinear optimization approaches, it covered both first order and second order gradient methods to achieve the fastest method for solving the collinearity equations nonlinearity problem.

The object space transformation parameters were successfully estimated (in terms of accuracy and number of iterations) by the Quasi-Newton approach with three ground control points (Sheta et al. 2012a). This leads to reducing the number of matches needed for estimating the parameters mentioned above, which will assess in the image matching approach for solving for the necessary points needed for convergence of the object space transformation parameters estimation problem.

Chapter Five: **INS/VISION MEASUREMENT FUSION USING EXTENDED KALMAN FILTER (EKF)**

This chapter introduces the general structure of EKF for UAV navigation based on INS and visual measurements in GPS-denied environments. The system applies the concept of photogrammetric resection to estimate the camera position which in turn can be used as an update to the IMU measurements in the navigation EKF. The camera position is estimated by finding the correspondences of common features exiting in the real-time images captured during flight mission and the geo-referenced images in the database of the area of flight. Then, the collinearity equations are used to estimate the position of the perspective center of the camera through the concept of photogrammetric resection. The object space transformation parameters, the output of the photogrammetric resection, are then used as an update to the INS in the navigation filter.

5.1 Introduction

UAV reliable navigation has been a subject for research for many years and still is. The work presented in this chapter is done in the context of UAV navigation systems. The most common navigation approaches for UAV applications are INS and GPS which can estimate the navigation parameters for the UAV in motion. However, both systems have their own limitations.. For example, INS alone is a dead-reckoning system which suffers from time dependent position error drift with time due to the integration of the acceleration and angular rate data for UAV navigation. On the other hand, GPS provides absolute and drift free position measurements provided that line of sight condition is realized between the GPS receiver and four or more satellites.

The combination of INS and GPS systems bound the navigation error drift from the INS system by the GPS updates. However, GPS-aided navigation systems are not always guaranteed to be reliable especially in GPS-denied environments, where the GPS signal can be jammed.

In order to overcome the main disadvantages of traditional methods (INS/GPS), vision-aided navigation systems are introduced to relieve the dependency on GPS in the integrated navigation system. Visual sensors have the advantages of low cost and weight compared to laser sensor. These advantages are crucial in the development of UAV navigation systems.

New advances in visual sensor solutions, in combination with traditional navigation sensors, are proving to be highly effective replacements to traditional IMU or GPS systems for many mission scenarios.

The primary objective of this chapter is to develop and implement a comprehensive, fast, real-time, and low cost VBN technique for UAVs navigation. This will be achieved by fusing the vision measurements as CUPT update through the EKF for the INS measurements by employing the residuals originating from the object space transformation parameters estimation as the error covariance matrix in the EKF implementation. The implementation will be done through simulated visual measurements combined with simulated GPS/INS trajectories at low altitude.

The proposed VBN approach is based on locating the correspondence points between a set of features in real-time captured images taken by the imaging sensor on the UAV and database images. These correspondence points between the geo-referenced database images and those captured in real-time are found by employing the fast modified SURF algorithm.

The chapter starts with the mathematical model for the collinearity equations and photogrammetry resection as the core of the camera pose estimation approach. Then, the general structure for the developed VBN system is introduced with detailed description for the INS error

states, the EKF, and fusing visual measurements with INS measurements for UAV navigation during GPS signal outages. Detailed description for the Spirent simulator, which was used for testing the developed methodology, is presented.

Finally, to test the developed VBN, simulated data set was used using the Spirent GNSS simulator to test the validity of the VBN through simulated GPS signal outages and compare the system performance to the simulator true trajectory.

5.2 Collinearity equations and photogrammetric resection

The relationship between the image points and the point features coordinate is described by the collinearity equations. In a VBN system, the position of a UAV can be estimated from the on-board navigation system or by using the conjugate correspondence points from the matching between the UAV real-time images and the images in the database. As described in chapter 4, the camera Exterior Orientation Parameters (EOP) $(\omega, \varphi, \kappa, X_{PC}, Y_{PC}, Z_{PC})$ can be estimated through the collinearity equations. These parameters define the relationship between the camera frame (c-frame) and the mapping frame (m-frame) shown in Figure 4.1. This is achieved using the proper optimization algorithm to ensure accurate and fast convergence of the parameters with the minimum required conjugate points necessary for the solution, as described in chapter 4.

Based on the collinearity equations 4.4, one of the basic problems of photogrammetry, named resection, can be solved. Space resection is a process which involves determining the position and orientation of a photograph, using measurements to features with known ground coordinates appearing on the photograph. The most common application is to aerial photogrammetry when the measured points are ground points, i.e. points which have known values in the object system of coordinates. Through the space resection, the EOP can be estimated when at least three GCPs are available.

5.3 Proposed vision and inertial fusion algorithm

The VBN algorithm will begin by building a library of classes for the geo-referenced satellite images of the area of interest. This library will be built on the SURF with descriptor length 36 (SURF-36) algorithm and will contain all descriptors for selected features that form the classes. Once a set of key points is detected in the real-time aerial images, they are matched with the set of SURF descriptors available in the database. This matching strategy is based on the Euclidean distance in descriptor space and is referred to as the similarity-threshold-based matching strategy.

Given conjugate correspondence points between the real-time images and the images from the database, the object-space transformation parameters can be estimated through the mathematical model of the collinearity equations using the proper optimization algorithm to ensure accurate and fast convergence for those parameters with the minimum required conjugate points necessary for convergence.

The vision aid for UAV VBN systems uses the image matching position to update the navigation- EKF, as shown in Figure 5.1. Rather than using the GPS positions to update the INS error states, the new proposed algorithm will add the image matching updates to that filter. The filter will then use two sets of observation to perform an update.

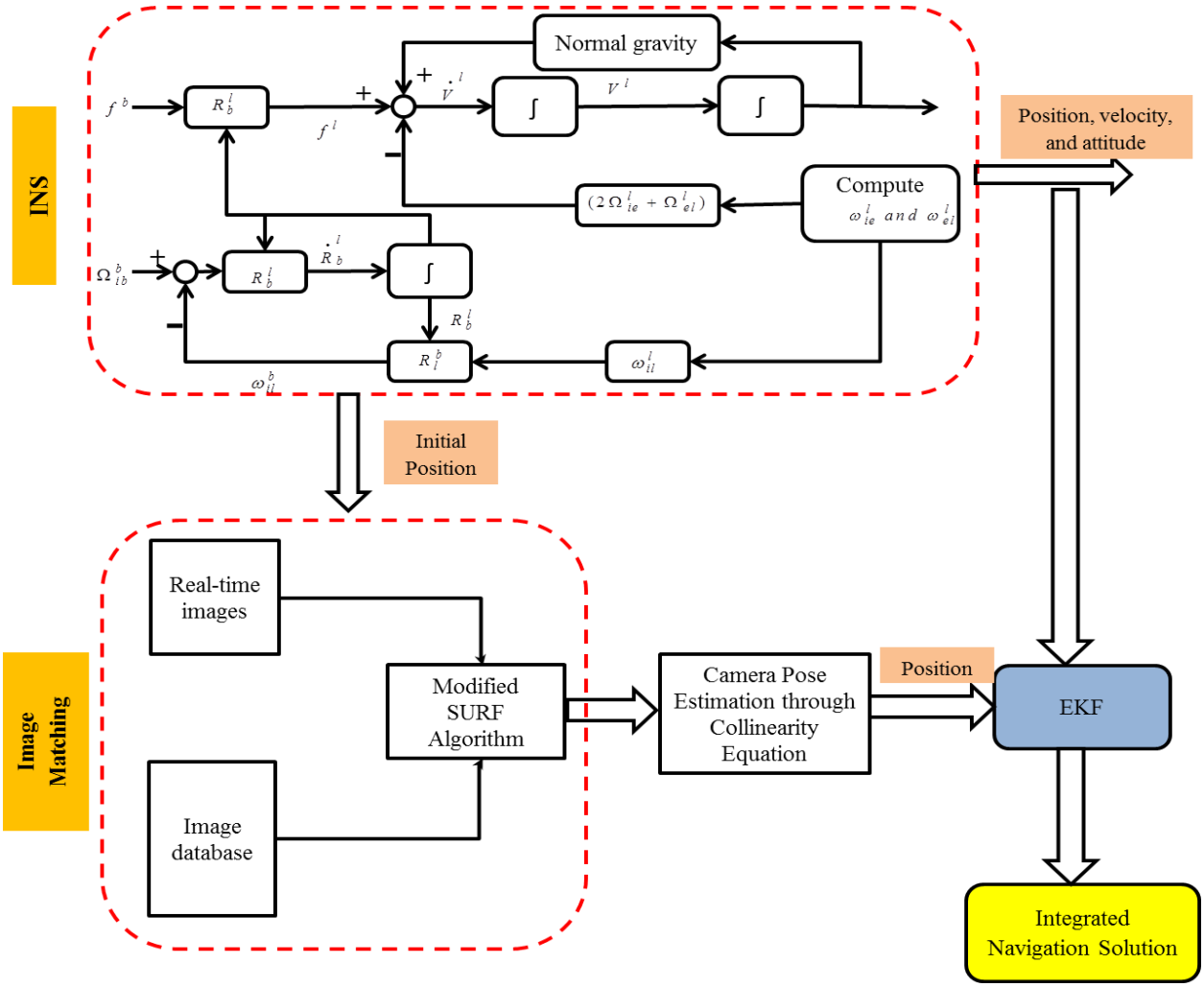


Figure 5.1: Vision aiding algorithm for UAV VBN

5.3.1 INS error states

The nonlinear state equations in chapter 2 describe a physical process, of relating the IMU measurement to the navigation states, by a deterministic dynamic system. Navigation parameters along a reference trajectory can be determined from the state equations using kinematic measurements. Due to sensor errors, solutions for the state equations in chapter 2 contain errors which can be deterministic or stochastic in nature. Thus, error models are required for analysis and estimation. Differential equations are used to describe the errors in dynamic systems because these errors are variable in time. Linearization of the nonlinear dynamic system is the most

common approach to derive a set of linear differential equations which define the error states of the dynamic system.

Recalling equation 2.29 that represents a system of nonlinear first-order differential equations (mechanization equations) then kinematic measurements are used to solve it to provide positions, velocities and attitudes. It can be rewritten using the form:

$$\dot{x}_1(t) = f_1\{(x_1(t), x_2(t); t)\} \qquad \textbf{5.1}$$

Where:

$f(x_1(t), x_2(t))$ are nonlinear functions, $x_1(t)$ are physical parameters of the dynamic system (representing the navigation state) and $x_2(t)$ $x_2(t)$ are the the input forcing functions (representing the sensor error models).Usually, the true values of the navigation state $x_1(t)$ are not known and an approximation of $x_1(t)$ can be computed, based on a reference trajectory, by integrating the sensors output with respect to time as follows:

$$\dot{\tilde{x}}_1(t) = f_1\left\{\left(\tilde{x}_1(t), x_2(t); t\right)\right\} = f_1\{(x_1(t) + dx_1(t), x_2(t); t)\} \qquad 5.2$$

Where:

$\tilde{x}_1$ is the approximation of the navigation states x_1.

$dx_1(t) = \tilde{x}_1(t) - x_1(t)$ is the error states.

Using a Taylor expansion to first order on equation 5.2, the linearized time derivative of the navigation error states will be:

$$\dot{dx}_1(t) = \dot{\tilde{x}}_1(t) - \dot{x}_1(t) = \frac{\partial f_1}{\partial x_1} dx_1(t) \qquad \textbf{5.3}$$

Equation 5.3 represents the navigation error state equations under the assumption that the input forcing functions $x_2(t)$ are errorless and can be written in a general form as:

$$\dot{dx}_1(t) = F_1(t) dx_1 \qquad \textbf{5.4}$$

Where F_1 is called the dynamic matrix.

If the input forcing functions $x_2(t)$ are not zero but are part of the sensor output, the navigation error state model in equation 5.4 is augmented by a set of error states to represent the random sensor noise. A set of stochastic differential equations are used to describe the sensor errors model as follows:

$$\dot{dx}_2(t) = F_2(t) dx_2(t) + G_2 w(t) \qquad \textbf{5.5}$$

Where:

G_2 is a coefficient matrix (usually called shape matrix).

$w(t)$ is system input white Gaussian noise.

The complete linearized INS error model is described by combining equations 5.4 and 5.5 as follows:

$$\begin{bmatrix} \dot{dx}_1(t) \\ \dot{dx}_2(t) \end{bmatrix} = \begin{bmatrix} F_1 & F_{12} \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} + \begin{bmatrix} 0 \\ G_2 \end{bmatrix} w(t) \qquad \textbf{5.6}$$

The general form for the linearized INS error model can be described as follows:

$$\dot{x}(t) = F(t) x(t) + G_{error} w(t) \qquad \textbf{5.7}$$

Where:

$x(t) = \begin{bmatrix} \dot{dx}_1(t) \\ \dot{dx}_2(t) \end{bmatrix}$ is the error states.

$F(t) = \begin{bmatrix} F_1 & F_{12} \\ 0 & F_2 \end{bmatrix}$ is the dynamic matrix.

$G_{error}w(t)$ describes the system noise.

The navigation error state vector $x(t)$ in the navigation solution produced by the INS is composed of the position, velocity, attitude, accelerometer bias, gyroscope bias, accelerometer scale factor, and gyroscope scale factor and is represented as a vector of 21 elements as shown below (Eun-Hwan and El-Sheimy 2004).

$$\boldsymbol{x(t)} = \begin{bmatrix} \boldsymbol{\delta P} \\ \boldsymbol{\delta V} \\ \boldsymbol{\psi} \\ \boldsymbol{a}^b \\ \boldsymbol{g}^b \\ \boldsymbol{a}^s \\ \boldsymbol{g}^s \end{bmatrix} \qquad \textbf{5.8}$$

Where:

δP is the position error vector.

δV is the velocity error vector.

ψ is the attitude error angles.

a^b is the accelerometer bias error.

g^b is the gyroscope bias error.

a^s is the accelerometer scale factor error.

g^s is the gyroscope scale factor error.

INS errors state (dx_2) can be categorized into three main types (El-Sheimy 2007): bias errors, scale factor errors, and noise, as shown in Figure 5.2.

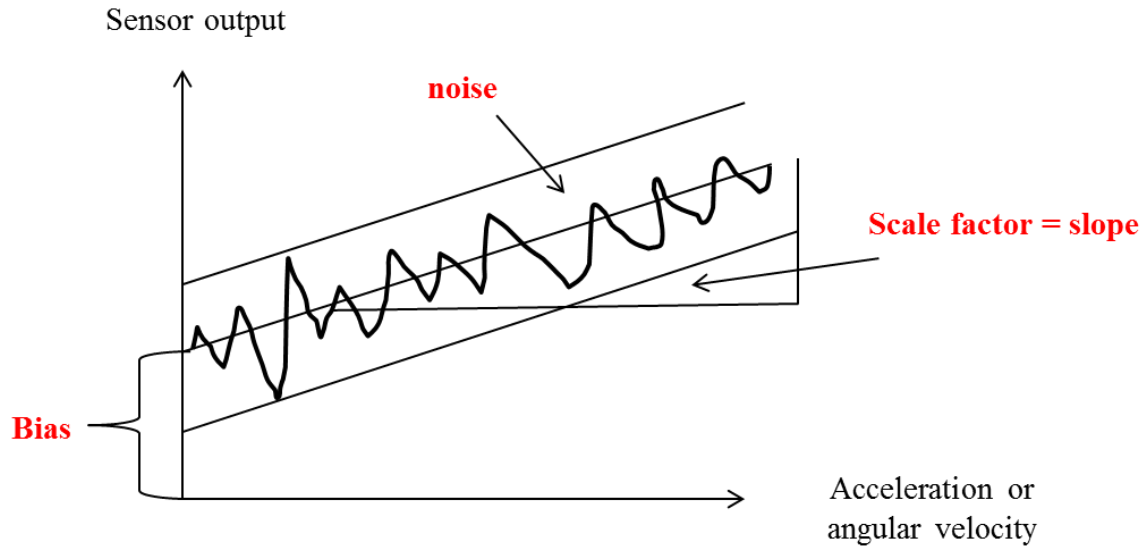


Figure 5.2: INS error types

The bias error includes two parts, as shown in Figure 5.3:

1. The bias offset which is a deterministic error that describes the error in the measurements of the inertial sensor (gyros and accelerometers).
2. The bias drift which is a stochastic part that describes the error accumulation of the INS sensor error with time.

The bias offset can be determined and removed through the proper calibration method where the bias drift is random and should be modeled as a stochastic process.

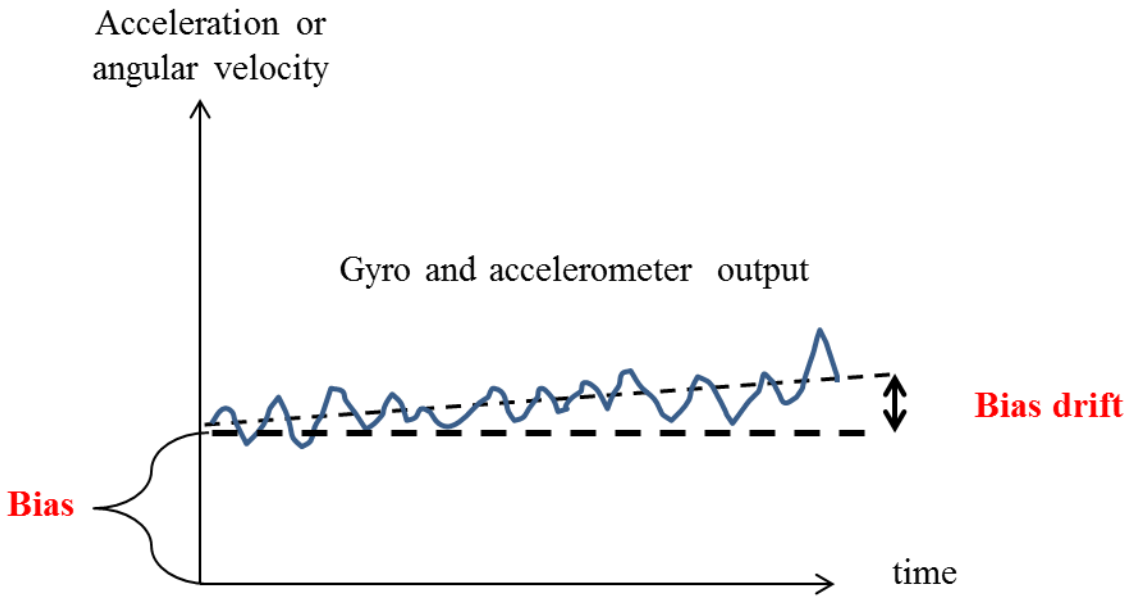


Figure 5.3: Bias offset and bias drift (El-Sheimy 2007)

The scale factor error is defined as the ratio between the measured sensor output and the true sensor measurement. It is a deterministic error and is determined and removed by the proper calibration method. Scale factor stability can be described as the variation of the scale factor with the temperature and is expressed in part per million (ppm).

As for noise error, it is a non-systematic error which cannot be removed using the deterministic model and is modeled by the stochastic process. The most common noise distribution is the white noise distribution, where the spectral density is the same at all frequencies.

The white noise process characteristics are zero mean and at constant power spectral density when stationary (Anderson and Moore 1979). As a stationary process, it is defined by its autocorrelation function specifications (Brown and Hwang 1992). For stationary random process, the autocorrelation function is defined as:

$$R_{bb}(\tau) = E\left[b(t)b(t+\tau)\right] \qquad \textbf{5.9}$$

where:

$R_{bb}(\tau)$ is the autocorrelation function of the signal *b*.

The signal *b* is the output sequence of the inertial sensor.

Given that the white noise has zero mean and constant power spectral density, the autocorrelation function of the white noise indicates zero correlation for all time lag values except at time lag = 0, as shown in Figure 5.4.

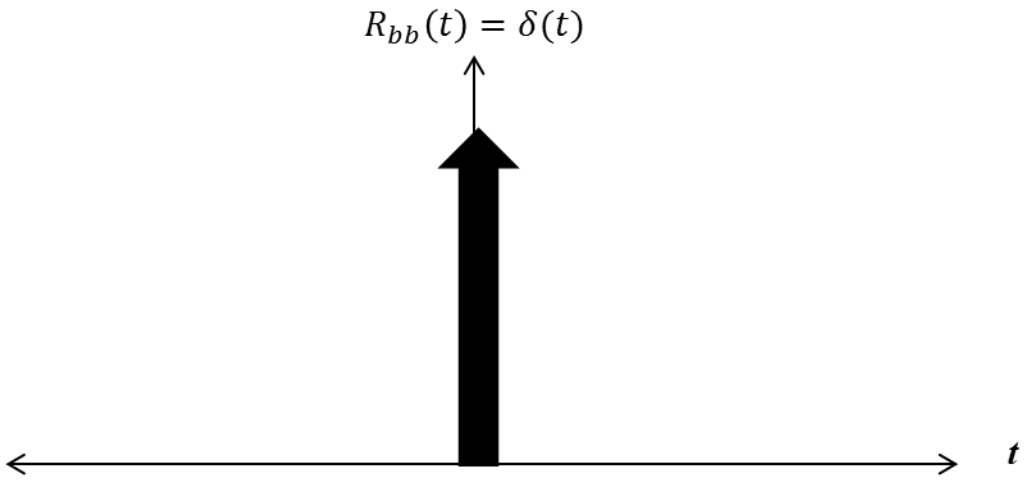


Figure 5.4: Autocorrelation function of white noise

The effect of the above INS errors on navigation parameters can be summarized as:

1. The accelerometer bias error introducing a linear error in velocity and a quadratic error in the position.
2. The gyro bias error introducing a quadratic error in velocity and a cubic error in position.

Therefore, for stand-alone low-cost INS, the aerial platform is unable to navigate correctly. Consequently, the low-cost INS will be aided by visual measurements for fast and accurate navigation.

5.3.2 Extended Kalman Filter (EKF)

The pose estimation process through image matching can be described as a filtering problem, where noisy observations of the interest points are utilized to estimate the pose of the aerial platform at time t.

The Kalman filter approach is considered one of the most efficient techniques for solving such problems. However, when the system models are nonlinear, like in the current case, the EKF is used. In this case, the EKF is established through linearizing such state models.

The observation model is described in Figure 5.5, where matches between visual features detected in the real-time captured image and the corresponding features in the geo-referenced database images are used as observation data.

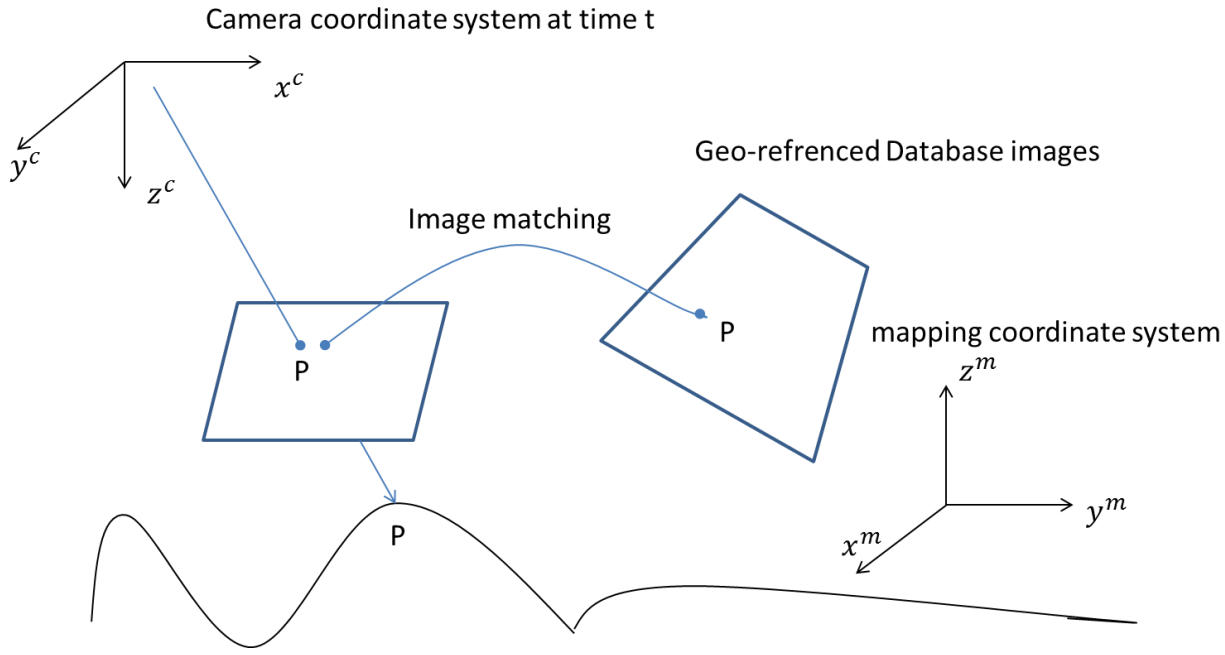


Figure 5.5: The observation model

The Kalman filter is introduced in the following according to (Brown and Hwang 1997). To derive the Kalman filter equations, a number of assumptions should be realized. The first assumption is discrete random process of the model given as:

$$x_{k+1} = \Phi_k x_k + W_k$$

$$E\{W_k\} = 0 \forall k \quad , E\left\{W_k W_j^T\right\} = \begin{cases} Q_k & j = k \\ 0 & j \neq k \end{cases} \qquad \textbf{5.10}$$

Where:

x is the state space vector.

Φ is the transition matrix which transforms initial state to its corresponding state at time k.

W is the process noise.

The state space vector x is related to the observations by:

$$u_k = H_k x_k + n_k$$

$$E\{n_k\} = 0 \quad \forall k \quad , E\left\{n_k n_j^T\right\} = \begin{cases} R_k & j = k \\ 0 & j \neq k \end{cases} \qquad \textbf{5.11}$$

Where:

u is the measurement vector or observations.

H is the design matrix connecting the measurements and the state vector.

n is the measurement noise.

The second assumption is that the initial estimate of the state vector at time t_k is known based on the process prior to t_k which is called a priori information and denoted as $\hat{x}_{k-1}$. When the observations u_k occur, the current estimate $\hat{x}_k$ can be improved by the priori information according to:

$$\hat{x}_k = \tilde{K}_k \hat{x}_{k-1} + K_k u_k \qquad \textbf{5.12}$$

Where:

K is the Kalman gain to be determined.

The error between the estimate $\hat{x}_{k-1}$ and the true value x_k is given by:

$$e_{k-1} = \hat{x}_{k-1} - x_k \qquad \textbf{5.13}$$

$$e_{k+1} = \hat{x}_{k+1} - x_k \qquad \textbf{5.14}$$

Using equations 5.11and 5.12 in equation 5.14 , the updated estimate error is given by:

$$\begin{aligned} e_{k+1} &= \tilde{K}_k \hat{x}_{k-1} + K_k u_k - x_k \\ &= \tilde{K}_k \left(e_{k-1} + x_k \right) + K_k \left(H_k x_k + n_k \right) - x_k \\ &= \left(\tilde{K}_k + K_k H_k - \mathrm{I} \right) x_k + \tilde{K}_k e_{k-1} - K_k n_k \end{aligned} \qquad \textbf{5.15}$$

Given $E\{n_k\} = 0$ and $E\{e_{k-1}\} = 0$, an unbiased estimate exists only if the following condition is realised:

$$\tilde{K}_k = \mathrm{I} - K_k H_k \qquad \textbf{5.16}$$

Therefore, the updated state space vector is given by:

$$\begin{aligned} \hat{x}_{k+1} &= \left(\mathrm{I} - K_k H_k \right) \hat{x}_{k-1} + K_k u_k \\ &= \hat{x}_{k-1} + K_k \left(u_k - H_k \hat{x}_{k-1} \right) \end{aligned} \qquad \textbf{5.17}$$

Comparing equations 5.17 and 5.11 , a new term called innovation can be defined as the difference between the current and predicted measurements,

$$\begin{aligned}\hat{n}_{k-1} &= u_k - H_k \hat{x}_{k-1} \\ &= u_k - \hat{u}_{k-1}\end{aligned} \qquad \textbf{5.18}$$

Therefore, equation 5.17 can be written as:

$$\hat{x}_{k+1} = \hat{x}_{k-1} + K_k \hat{n}_{k-1} \qquad \textbf{5.19}$$

Therefore, corrections to the predicted measurements are used to correct the predicted state space vector, which is the main idea of the Kalman filter. The Kalman gain K is used to determine the effect of such measurements corrections. The Kalman gain is defined as the mapping of measurements error and state space vector errors, as shown below:

$$\hat{x}_{k+1} - \hat{x}_{k-1} = K_k \left(u_k - \hat{u}_{k-1} \right) \qquad \textbf{5.20}$$

The error covariance matrix of the updated estimate is given by:

$$P_{k+1} = E\left\{ \hat{e}_{k+1} \hat{e}_{k+1}^T \right\} = E\left\{ \left(\hat{x}_{k+1} - x_k \right) \left(\hat{x}_{k+1} - x_k \right)^T \right\} \qquad \textbf{5.21}$$

The error between the estimate of the updated state space and the current state space can be given as:

$$\begin{aligned}\hat{x}_{k+1} - x_k &= \hat{x}_{k-1} + K_k \left(u_k - H_k \hat{x}_{k-1} \right) - x_k \\ &= \left(\hat{x}_{k-1} - x_k \right) + K_k \left(H_k x_k + n_k - H_k \hat{x}_{k-1} \right) \\ &= \left(\hat{x}_{k-1} - x_k \right) + K_k H_k \left(\hat{x}_{k-1} - x_k \right) + K_k n_k \\ &= \left(\mathrm{I} - K_k H_k \right) e_{k-1} + K_k n_k\end{aligned} \qquad \textbf{5.22}$$

Therefore, the error covariance matrix P_{k+1} is given by:

$$
\begin{aligned}
P_{k+1} &= E\left\{\left(\hat{x}_{k+1} - x_k\right)\left(\hat{x}_{k+1} - x_k\right)^T\right\} \\
&= E\left\{\left[\left(\mathrm{I} - K_k H_k\right)e_{k-1} + K_k n_k\right]\left[\left(\mathrm{I} - K_k H_k\right)e_{k-1} + K_k n_k\right]^T\right\} \\
&= \left(\mathrm{I} - K_k H_k\right)E\left\{e_{k-1}e_{k-1}^T\right\}\left(\mathrm{I} - K_k H_k\right)^T + K_k E\left\{n_k n_k^T\right\}K_k^T \\
&+ \left(\mathrm{I} - K_k H_k\right)E\left\{e_{k-1}n_k^T\right\}K_k^T + K_k E\left\{n_k e_{k-1}^T\right\}\left(\mathrm{I} - K_k H_k\right)^T
\end{aligned}
\qquad \textbf{5.23}
$$

Given the following two conditions:

1. Uncorrelated measurement errors with the estimation error, i.e.

   $E\left\{e_{k-1}n_k^T\right\} = E\left\{n_k e_{k-1}^T\right\} = 0$.

2. $P_{k-1} = E\left\{e_{k-1}e_{k-1}^T\right\}$ and $R_k = E\left\{n_k e_{k-1}^T\right\}$

The error covariance matrix is given as:

$$P_{k+1} = (\mathrm{I} - K_k H_k)P_{k-1}\left(\mathrm{I} - K_k H_k\right)^T + K_k R_k K_k^T \qquad \textbf{5.24}$$

To achieve optimality condition in the Kalman filter, the minimum mean squared error condition is used. Thus:

$$f(e) = e^T e = minimum \qquad \textbf{5.25}$$

The sum of the mean squared errors in the estimates of x can be computed by minimizing the trace of error covariance matrix P_{k+1}. Therefore,

$$\frac{\partial trace\left(P_{k+1}\right)}{\partial K_k} = -2(\mathrm{I} - K_k H_k)P_{k-1}H_k^T + 2K_k R_k K_k^T = 0 \qquad \textbf{5.26}$$

Thus, the Kalman gain is given by:

$$K_k = P_{k-1}H_k^T\left(H_k P_{k-1}H_k^T + R_k\right)^{-1} \qquad \textbf{5.27}$$

The state space vector estimates and the error covariance matrix can be computed using equations 5.14, 5.24, and 5.27. To compute the update equations, the state space vector and the error covariance matrix should be projected to the observation time. Therefore, using equation 5.10 and ignoring the noise effect,

$$\hat{x}_{k+1} = \Phi_k \hat{x}_k \qquad \textbf{5.28}$$

To compute the error covariance matrix, the priori error is first computed as:

$$\begin{aligned} e_k &= x_{k+1} - \hat{x}_k \\ &= \left(\Phi_k x_k + w_k\right) - \Phi_k \hat{x}_k \\ &= \Phi_k e_k + w_k \end{aligned} \qquad \textbf{5.29}$$

Since w_k is the process noise for a step ahead of t_k, hence, the cross correlation between w_k and e_k equals zero. Therefore, the error covariance matrix is computed as:

$$\begin{aligned} P_k &= E\left\{e_{k-1}e_{k-1}^T\right\} = E\left\{\left(\Phi_{k-1}e_{k-1} + w_{k-1}\right)\left(\Phi_{k-1}e_{k-1} + w_{k-1}\right)^T\right\} \\ &= \Phi_{k-1}P_{k-1}\Phi_{k-1}^T + Q_{k-1} \end{aligned} \qquad \textbf{5.30}$$

5.3.2.1 Nonlinear observation model

Suppose the state of the UAV at time t is x_t and the observations coming from the vision measurements are u_t. The model describing both the system state and observations can be represented as:

State model: $$x_{t+1} = F_t(x_t) + W_t \qquad \textbf{5.31}$$

Observation model: $$u_t = H_t(x_t) + n_t \qquad \textbf{5.32}$$

Where $F_t(\cdot)$ and $H(\cdot)$ are nonlinear functions.

W_t, and n_t are the model and observation noise (Zhang et al., 2010).

The covariance matrix of the system noise is represented as (El-Sheimy 2007):

$$E\left[W_k W_k^t\right] = Q_k \qquad \textbf{5.33}$$

The covariance matrix of the measurement noise is represented as (El-Sheimy 2007):

$$E\left[n_k n_k^t\right] = R_k \qquad \textbf{5.34}$$

5.3.3 Coordinate UpdaTe (CUPT) for INS Aiding

The position measurements from camera perspective center estimation using collinearity equations are merged with the INS measurements in the Kalman filter using state observation equations. Since both INS position and image derived position measurements use the same coordinate frame, then the CUPT equation is a straight parameter equivalency as follows:

$$\begin{aligned} u_1 &= u_{image} = H_1 x + n_1 \\ u_{image} &= Position_{INS} - Position_{image} \end{aligned} \qquad \textbf{5.35}$$

Where:

$$H_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Prediction stage:

The Kalman filter will predict the states from epoch k-1 to epoch k using the transition matrix and the covariance matrix for system noise. This will estimate the states *(x)* and their covariance matrix *(P)*. The following equations are identical to any GPS/INS loosely coupled integrated filter.

$$x^{(0)} = \Phi_{k-1,k}.x_{k-1}$$
$$P^{(0)} = \Phi_{k-1,k}P_{k-1}\Phi_{k-1,k}^{T} + Q_{k-1} \qquad \textbf{5.36}$$

Updates stage:

Before predicting the second epoch, the states will undergo an update by the position measurements (image matched positions) and their covariance matrix (R_1 obtained from the residuals originating from the object space transformation parameters estimation).

$$K_1 = P^{(0)}H_1^T(H_1P^{(0)}H_1^T + R_1)^{-1}$$
$$x^{(1)} = x^{(0)} + K_1(Z_1 - H_1x^{(0)}) \qquad \textbf{5.37}$$
$$P^{(1)} = (I - K_1H_1)P^{(0)}$$

Operation of the proposed EKF is shown in Figure 5.6, where the EKF predict up to the time of CUPT and before the measurement update is done.

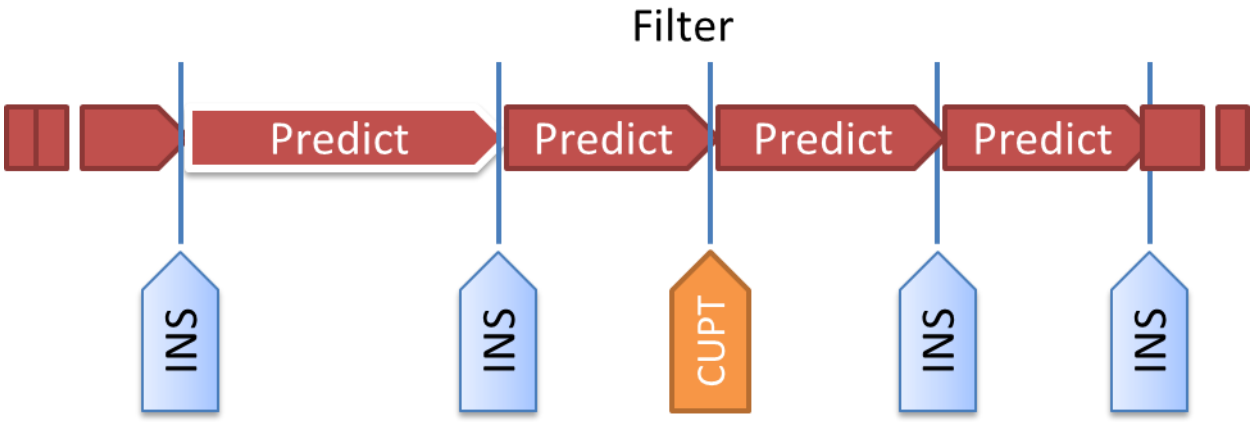


Figure 5.6: Operation of the proposed EKF

5.4 Data sets and experiments

To test the developed VBN, a simulated data set was used. The simulation was based on a trajectory over the Vancouver area as shown in Figure 5.7. Four data sets are included in the test.

The first data set was collected during the flight test with vertical camera axis as a representative of the database images and the second data set was a rotated and scaled image as representation of the real-time images captured from the UAV. These two data sets represent the visual measurements to the developed VBN system.

The camera specification and flight information for the data set of images taken from the Vancouver area are found in Figure 5.7.

The third data set is the simulated INS measurements from the Spirent GSS8000 and SimInertial GNSS/INS simulators. The fourth data set is the GPS measurements during the test which is simulated from the Spirent GSS8000 and SimInertial GNSS/INS simulators.

The validity of the developed VBN system is tested through simulated GPS signal outages and the system performance is compared to the simulator true trajectory.

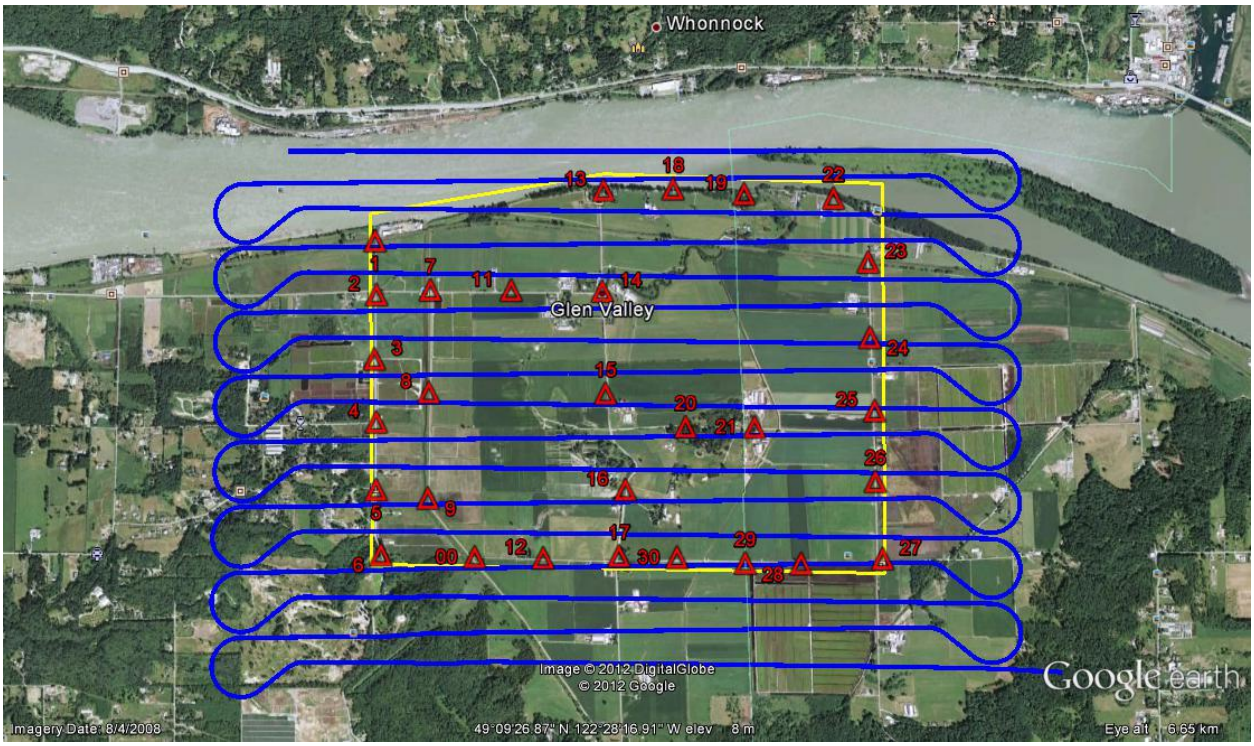


Figure 5.7: Flight trajectory with red triangles indicating GCP

The input to the EKF was obtained from simulated measurements using the Spirent GSS8000 and SimInertial GNSS/INS simulators to generate the required GPS and INS data. The generated scenario was of the flight path for aircraft flying over the flight test area coverage. The simulator then provided IMU measurements with an output rate of 100 Hz that was synchronized with the GPS RF signals. Figure 5.8 (Www.Spirent.com) shows a schematic of the Spirent GSS8000 and SimInertial GNSS/INS simulators data flow while Figure 5.9 shows a picture of the system's hardware.

The SimGEN PC models the test trajectory for the UAV flight for the Vancouver area described in Figure 5.7. The simulated UAV flight data is transferred to the SimInertial via Ethernet. The SimInertial system simulates inertial sensor outputs while in the meantime simulating GPS RF signals thereby provide simulated GPS/INS measurements to test the developed VBN system.

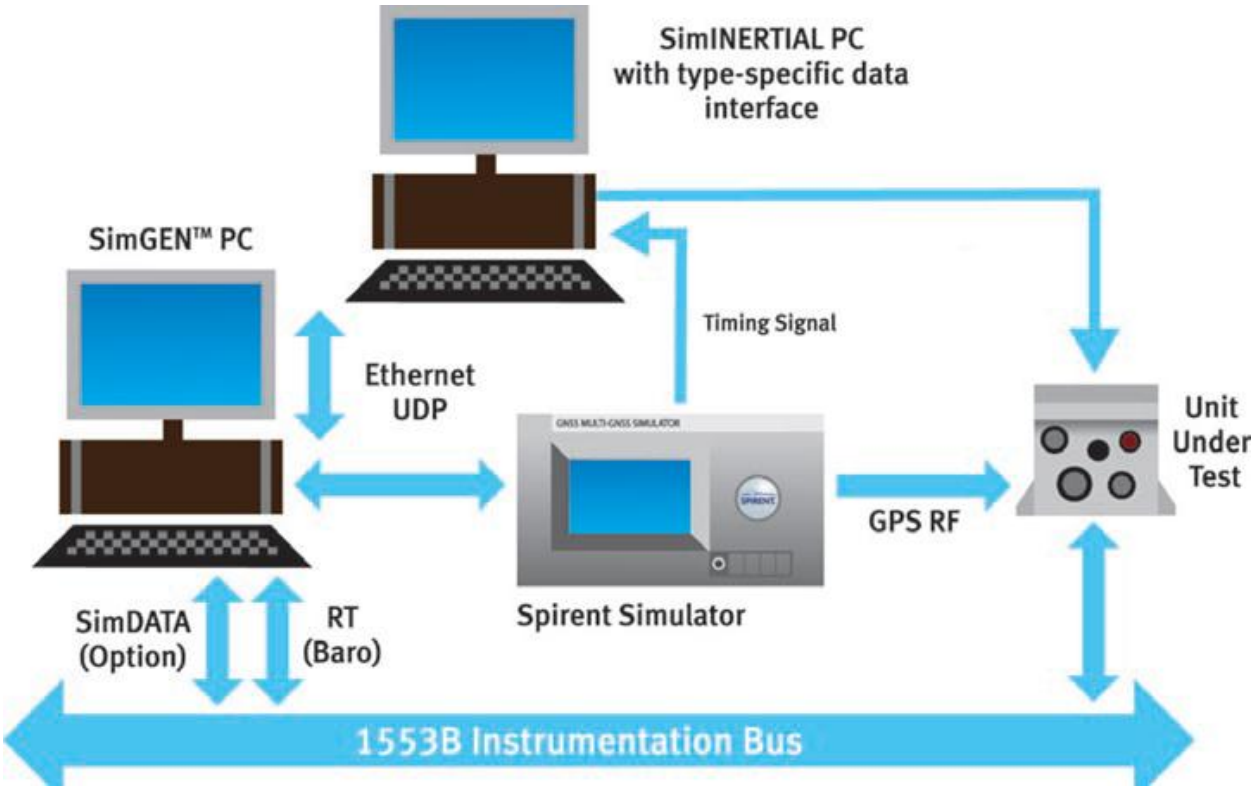


Figure 5.8: Simulator block diagram (Www.Spirent.com)

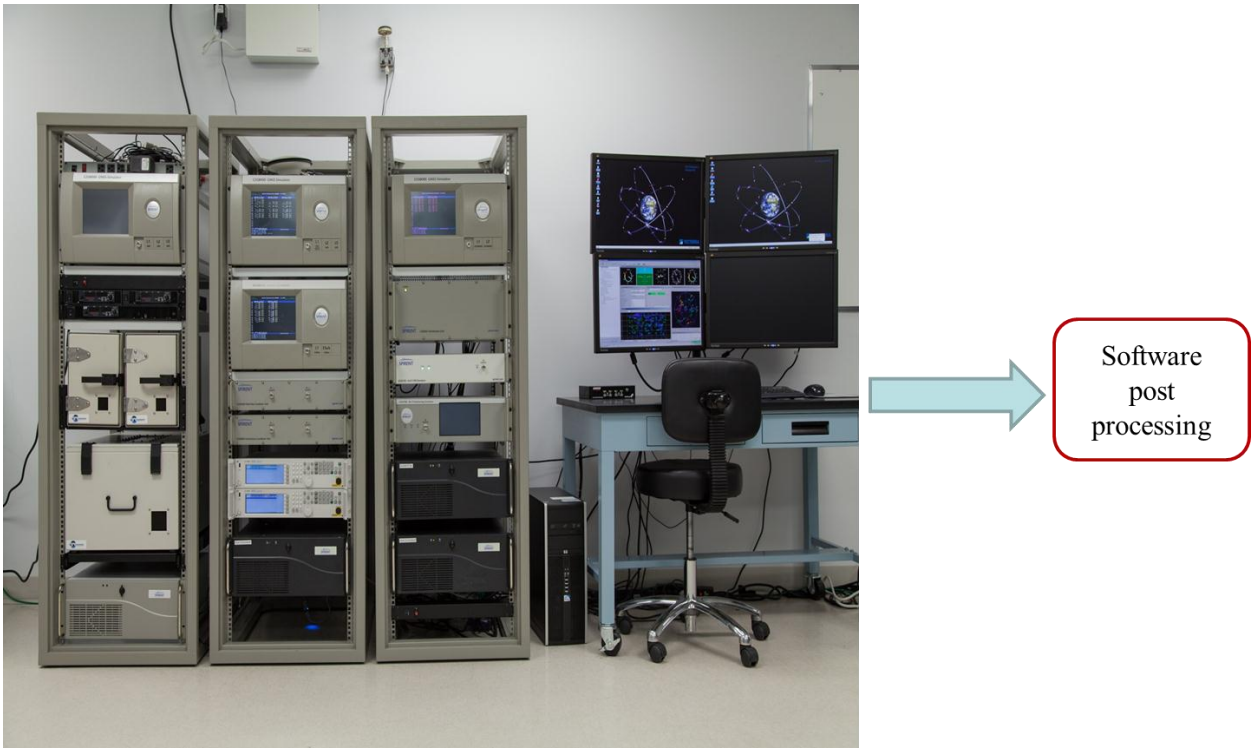


Figure 5.9: Hardware configuration for H/W simulator Spirent GSS 8000 used for flight scenario generation and testing.

The main test to assess the proposed VBN was conducted using simulated GPS signals, which were processed with simulated IMU data processing. The proposed algorithm was first tested using the previously described simulator hardware. The simulator is capable of providing IMU measurements with a controllable output rate and perfectly synchronized with the GPS RF signals. According to the type of application adopted in this work, the IMU simulated measurements is collected using NATO StanAg-4572 via the supplied RS422 card. A bias error was introduced in the simulated IMU data according to the following table:

Table 5-1: Introduced error to the simulated IMU data

Error source	Gyro bias	Accelerometer bias
Units	rad/sec	g
	0.00003	0.00003

IMU simulated measurements are processed to calculate the INS solution through the strap-down mechanization process described in chapter 2 section 2.4 as shown in Figure 5.10.

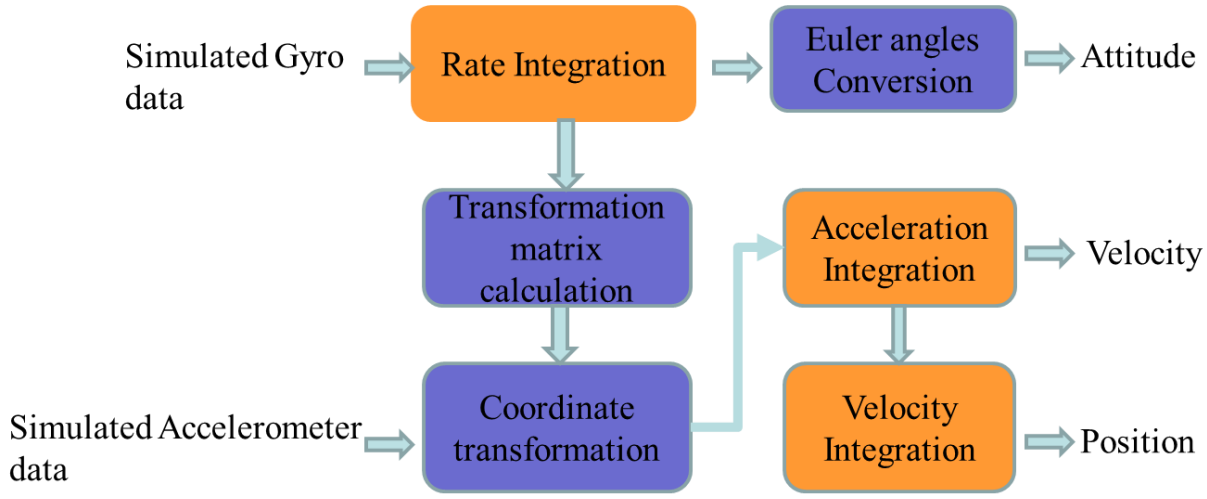


Figure 5.10: IMU mechanization block diagram

The simulated IMU and GPS data are then loosely coupled integrated through the developed EKF to provide a corrected estimate of the INS solution. The tests were conducted with neglecting the boresight and distortion parameters.

To test the effect of a GPS signal outage on the system performance, GPS signal outage was introduced on the GPS signal at specific times for 55 and 61 seconds in specific lines during the flight path. During this outage, the UAV travelled approximately 3 km (100 knots=51.44 m/sec x

60 sec = 3086 m). The navigation solution was then computed to determine the effectiveness of the VBN during GPS signal outages.

As a result of the GPS signal outage in lines 10 and 13 (as shown in Figure 5.11), the INS stand-alone navigation solution experience positional drift, as shown in Figure 5.12, at the end of the GPS signal outages with the values listed in Table 5-2.

Table 5-2: Maximum position error for navigation solution during GPS signal outage of 60 seconds

Maximum East Position Error (m)	Maximum North Position Error (m)	Maximum Up Position Error (m)
157.601	200.603	10.5766

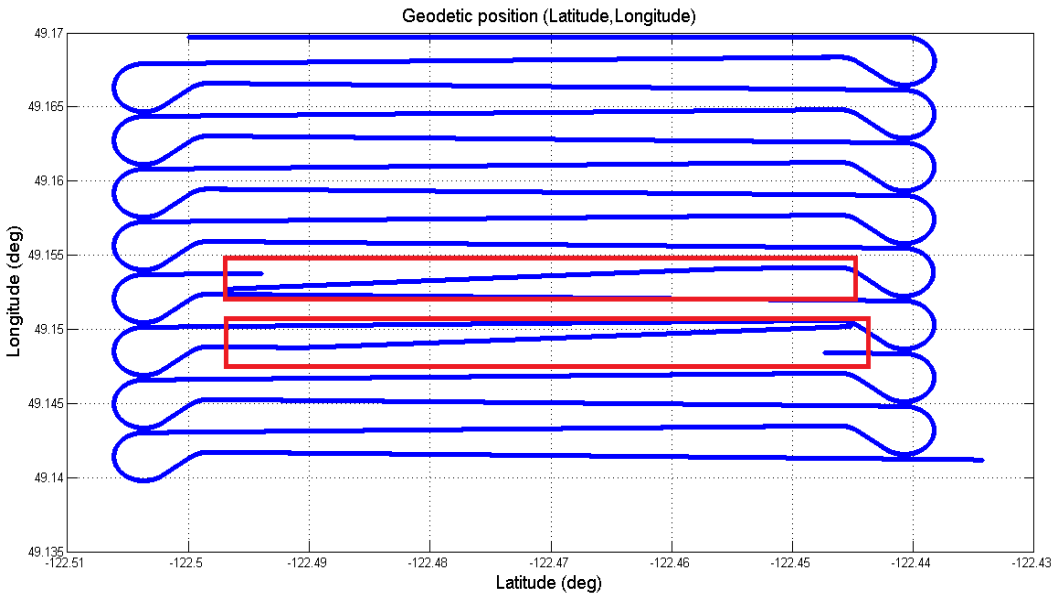


Figure 5.11: Navigation solution with introduced GPS signal outages in the simulated flight data

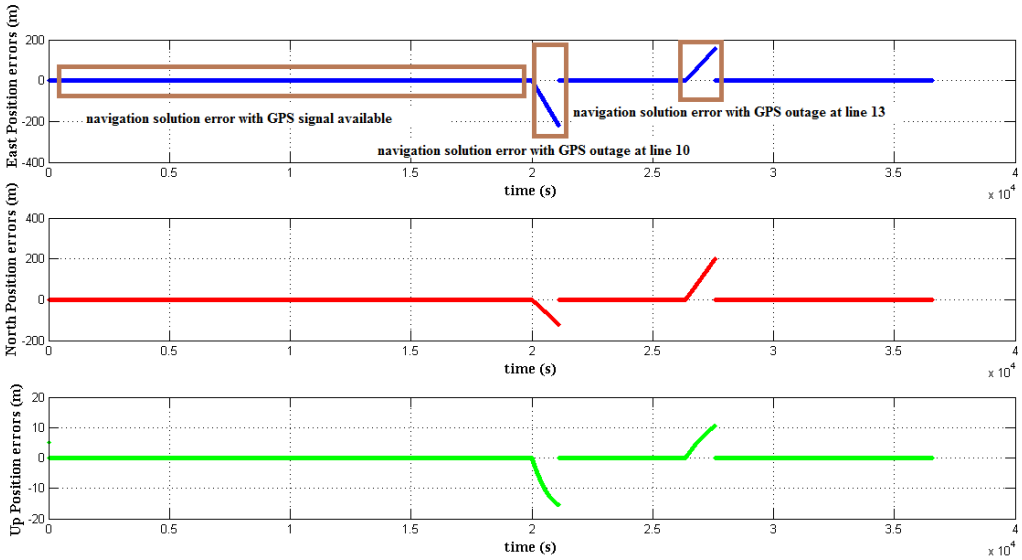


Figure 5.12: East, North, and Up position errors during GPS signal outage

Correspondences between real-time images captured during flight mission at Line 13 (tagged with red rectangle in Figure 5.13) and geo-referenced images in the database are found. Then, the collinearity equations are used to estimate the position of the perspective center of the camera through the concept of photogrammetric resection. The object space transformation parameters, the output of the photogrammetric resection, are then used as the update for the INS kalman for position updates. The navigation solution was improved as shown in Figure 5.13, Figure 5.14, and Figure 5.15, and Table 5-3.

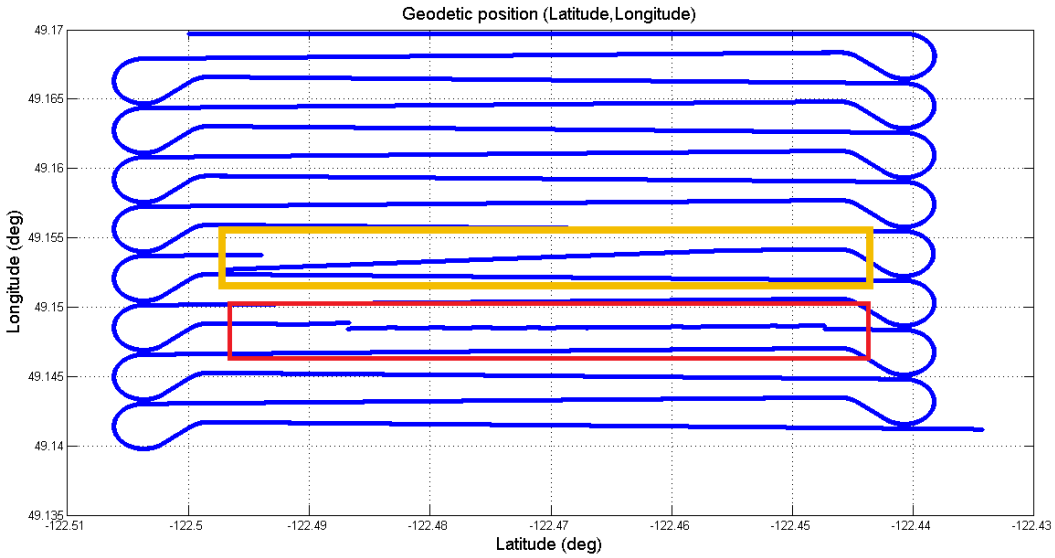


Figure 5.13: Navigation solution with the image updates to line 13 tagged with red rectangle

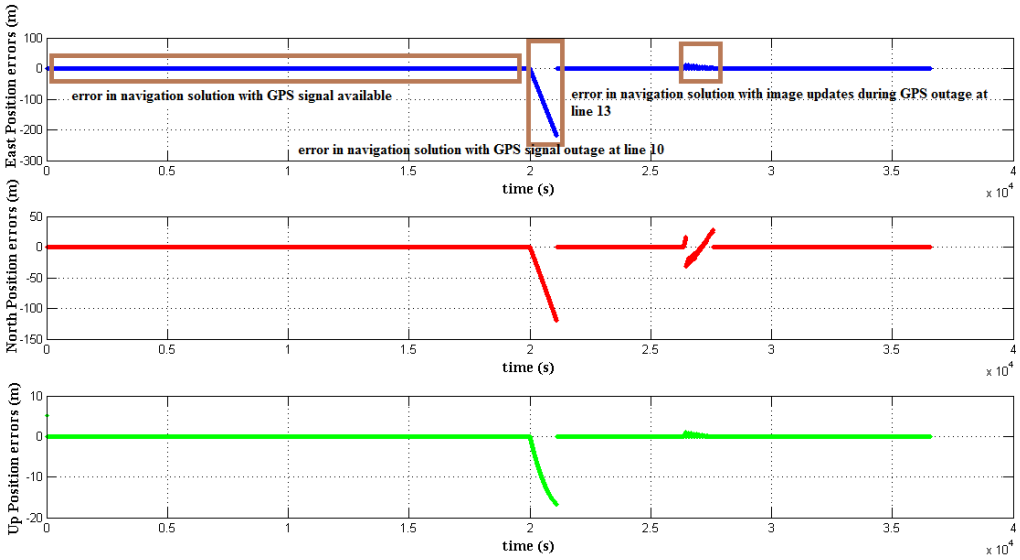


Figure 5.14 : East, North, and Up position errors after image updates

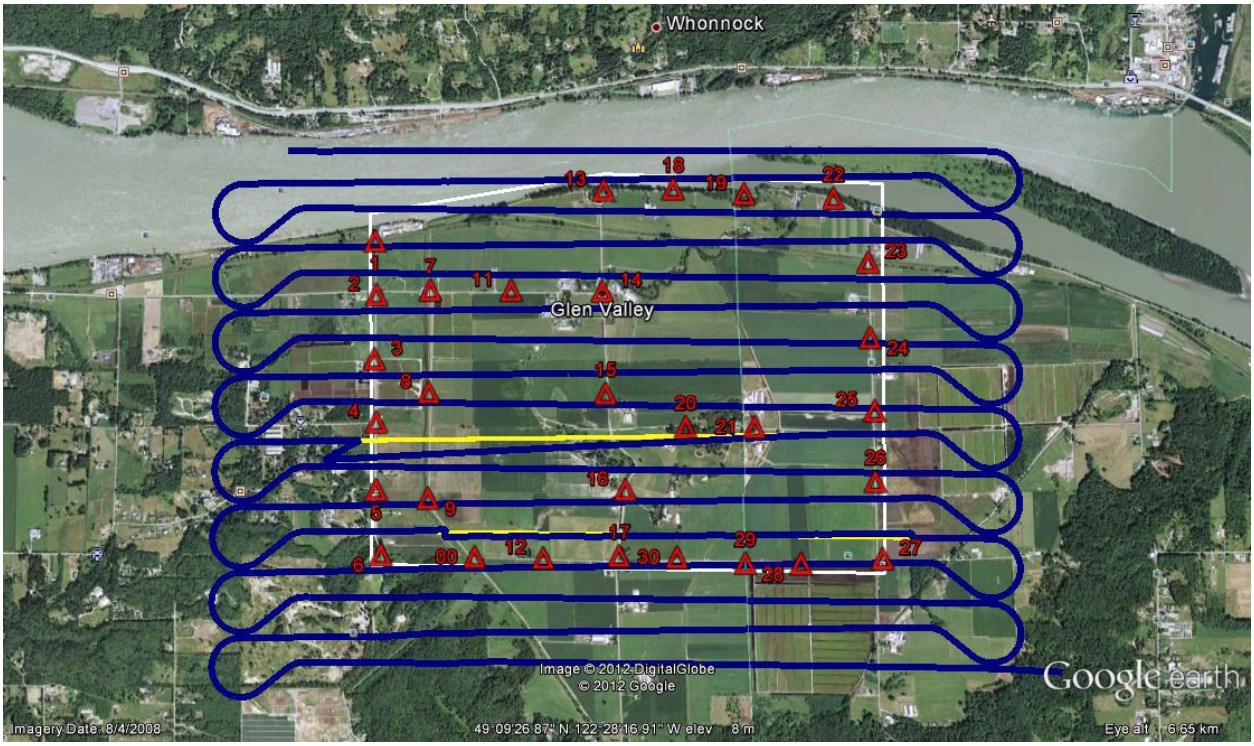


Figure 5.15: Navigation solution (blue) and truth flight path (yellow)

Table 5-3: Maximum position error for navigation solution after image update

Maximum East Position Error (m)	Maximum North Position Error (m)	Maximum Up Position Error (m)
13.0381	28.9542	5.0829

5.5 Summary

This chapter introduced and tested the developed VBN system for UAV navigation during simulated GPS signal outages. The developed VBN system is based on detecting the matches between real-time images captured during the UAV flight mission and geo-referenced images in the database. Then, the camera perspective center is estimated through the concept of photogrammetry resection via the collinearity equations model. The object space transformation

parameters, which is the output of the photogrammetric resection, are used as the update for the INS kalman for position updates A navigation solution with stand-alone INS solutions during a GPS outage showed poor performance resulting in an east position error of 157.601m, a north position error of 200.603m, and an upward position error of 10.74 m. This was expected due to the INS errors explained in section 5.3.1 . The proposed vision-based updates to the INS measurements offered improvements to the navigation solution as shown in Table 5-3.

Chapter Six: **CONCLUSIONS AND FUTURE WORK**

The thesis investigates the design issues related to the implementation of VBN methodology using imaging and inertial system for navigation applications of UAV during GPS signal blockage. As shown in Figure 6.1 and Figure 6.2, the proposed approach will enable the UAV to navigate to the desired destination during GPS signal outage. The main conclusions derived from the thesis, and recommendations for future work are discussed in this chapter.

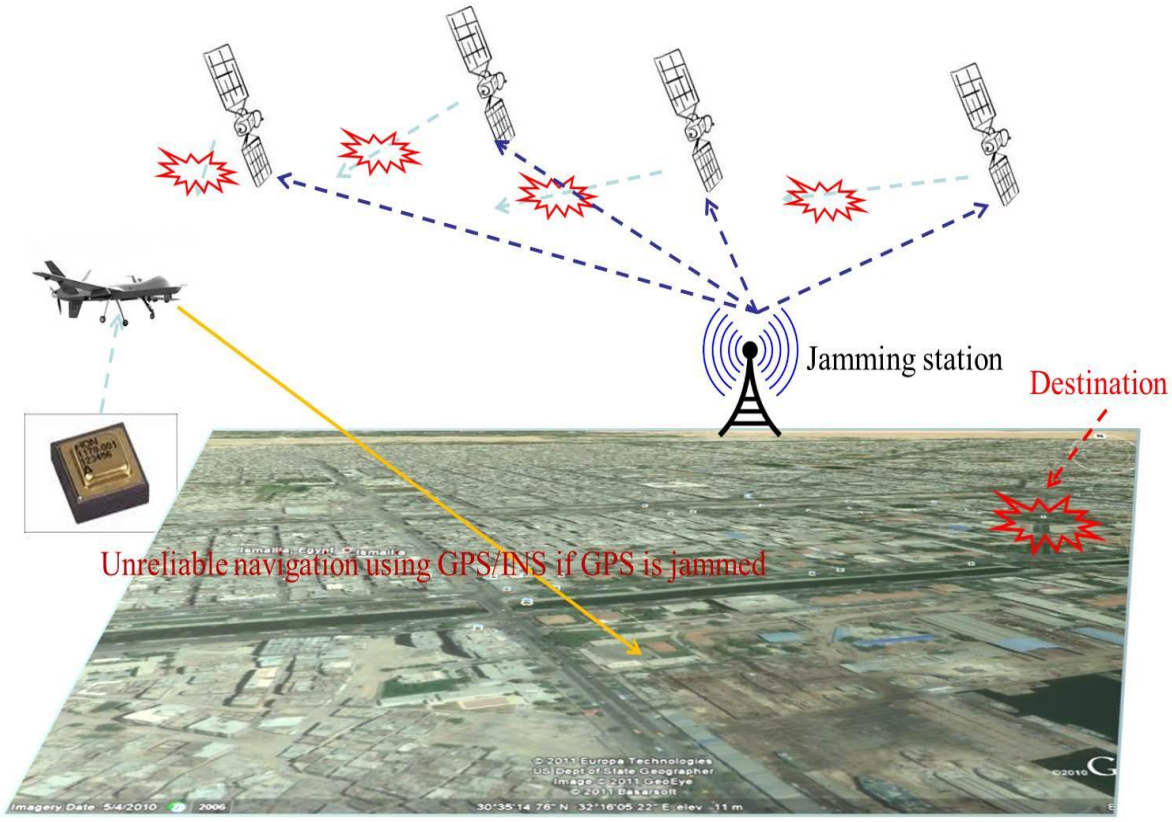


Figure 6.1: UAV navigation in GPS signal outage

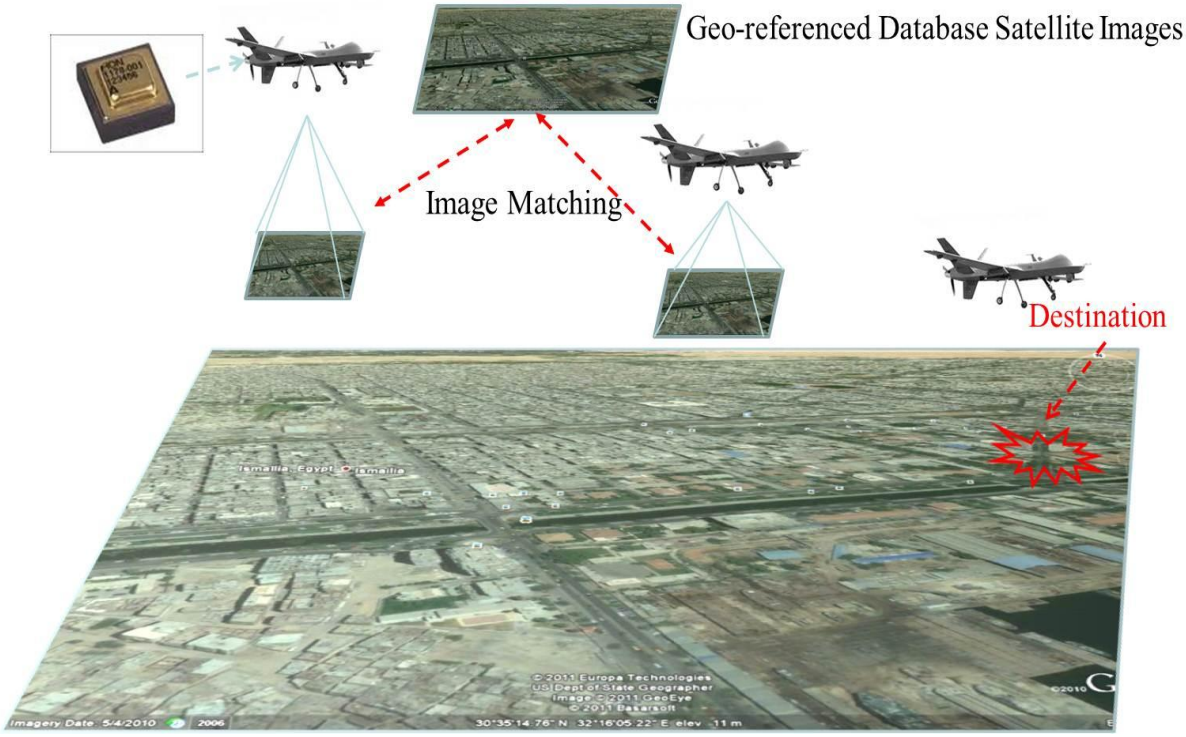


Figure 6.2: UAV VBN proposed approach

6.1 Contributions

The UAV industry is growing rapidly in an attempt to serve both military and commercial applications. A crucial aspect in the development of UAVs is the reduction of navigational sensor costs while maintaining accurate estimation of the UAV's navigation state. Advances in vision sensor and inertial navigation sensors is a promising solution for navigating UAVs in GPS denied missions.

This thesis contributes to the UAV research through the development of VBN approach for fast and accurate navigation in three main levels:

Image matching contribution: fast implementation using modified SURF-36 to meet real-time operation of UAV VBN with accurate pre-surveyed geo-referenced database images.

The image matching contribution includes:

1. Introducing and testing modified SURF-36 as the matching algorithm for VBN depending on a lower number of interest point matches between real-time captured image and images in the database.
2. Investigating and implementing a different samples count in the sub-divisions of the different types of SURF algorithm (SURF 36, SURF 64, and SURF 128) to test the effect of the number of samples in each subdivision on the accuracy of the matching algorithm. Results, clearly, showed that increasing number of samples in each subdivision is effective in the matching algorithm with high repeatability score up to 99%. This topic has not been previously investigated.
3. Introducing and implementation of the proposed algorithms on CUDA platform which showed a fast implementation employing of the shelf graphics card. Implementation using CUDA satisfies the real-time processing requirements, while keeping the high accuracy which is suitable for real-time application such as VBN.

Object space transformation parameters nonlinear optimization: Given the conjugate matched points between the real-time images and the images in the database, estimating the six transformation parameters (three for orientation and three for translation) for the Collinearity equations should be evaluated quickly and accurately. Linearization of the nonlinear mathematical model requires repeated computations to improve the initial values assigned to the transformation parameters. Although, in cases of near-vertical photography, ω and φ are usually small values and κ can be estimated and given an initial value. This situation is not usually the case, whereby the aerial platform will lead to a general photography with large orientation angle. Therefore, the contribution in this area includes:

1. Improving the performance of the algorithm, computational load prospective, used to solve for the transformation parameters starting with different levels of inaccuracies in the initial approximations.
2. Estimating the transformation parameters through the nonlinear model fitting using nonlinear optimization approaches.

INS/Vision Measurement Fusion using EKF: the contribution in this area is in the development of vision measurements to the update EKF as CUPT update for the INS measurements. In this case, the residuals originating from the image matching will act as the error covariance matrix in the EKF implementation and developing a simulation software package that can be used in testing and assessing the above mentioned parameters and UAV navigation.

6.2 Conclusions

The objectives of the thesis listed in Chapter 1 are to develop and implement a fast real-time low cost VBN methodology for UAV. The implementation was accomplished through simulated visual measurements combined with simulated GPS/INS paths at low altitude.

The thesis had met these objectives, whereby a complete package for UAV navigation based on visual measurements and emulates the processing onboard in real-time has been implemented and evaluated for UAV navigation.

The conclusions for the proposed VBN approach architecture are listed below:

Image matching proposed algorithm implementation

Modified SURF 36 algorithm has been developed and implemented to benefit from fast implementation while the accuracy of the matching process will depend on the accurate pre-surveyed geo-referenced database images and the number of samples in each sub-division in the

SURF 36 algorithm. Test results of the proposed image matching algorithm lead to the following conclusions:

1. The proposed modified SURF algorithm effect on the navigation solution showed an improvement in the navigation solution with position update from vision measurements compared to the navigation solution during GPS signal outage with 91.72% in the east position error and 85.56% in the north position error.
2. The proposed algorithm in this thesis has been implemented on CUDA platform which showed a fast implementation employing on the shelf graphics card. Implementation using CUDA satisfies the real-time processing requirements, while keeping the high accuracy which is suitable for real-time application such as VBN.
3. The overall speedup after the GPU CUDA implementation for 1280×960 image size compared to the CPU implementation is 99.63%.

Nonlinear optimization algorithms

A robust scale and rotation invariant image matching algorithm is vital for UAV VBN, where matches between an existing geo-referenced database images and the real-time captured images are used to georeference (i.e. to estimate the six transformation parameters - three rotation and three translation) the real-time captured image from the UAV through the collinearity equations. The georeferencing information is then used in aiding the INS integration Kalman filter as Coordinate UPdaTe (CUPT). It is critical for the collinearity equations to use the proper optimization algorithm to ensure accurate and fast convergence for georeferencing parameters with the minimum required conjugate points necessary for convergence. Fast convergence to a global minimum will require nonlinear approach to overcome the high degree of nonlinearity that exist in case of having large oblique images (i.e. large rotation angles).

Five different nonlinear least squares methods were presented for estimating the transformation parameters. Four gradient based nonlinear least squares methods (Trust region, Trust region dogleg algorithm, Levenberg-Marquardt, and Quasi-Newton line search method) and one non-gradient method (Nelder-Mead simplex direct search) is employed for the six transformation parameters estimation process.

Test results of the proposed nonlinear optimization algorithms lead to the following conclusions:

1. Failure of the traditional least squares approaches to estimate the georeferencing parameters, because of the expected nonlinearity of the mathematical model.
2. The research was done on simulated data and the results showed that the Nelder-Mead method has failed because of its dependency on the objective function without any derivative information.
3. The tested gradient methods succeeded in converging to the relative optimal solution of the georeferencing parameters. However, in trust region methods, the number of iterations was more than Levenberg-Marquardt because of the necessity for evaluating the local minimum to ensure if it is the global one or not in each iteration step.
4. As for the Levenberg-Marquardt method, which is considered as a modified Gauss-Newton algorithm, employing the trust region approach where a scalar is introduced to assess the choice of the magnitude and the direction of the descent. This scalar determines whether the Gauss-Newton method direction or the steepest descent method direction will be used as an adaptive approach for both linear and nonlinear mathematical models and it successfully converged and achieved the relative optimum solution.

5. The object space transformation parameters were successfully estimated only by Quasi-Newton line search method due to its dependence on the Hessian matrix of the Collinearity equations.

INS/Vision Measurement Fusion using EKF

The system applies the concept of photogrammetric resection to update the IMU measurements, which is achieved by finding correspondences between the real-time images captured during flight mission and the geo-referenced images in the data base. Then, the collinearity equations are used to estimate the position of the perspective center of the camera through the concept of photogrammetric resection. The object space transformation parameters, the output of the photogrammetric resection, are then used as the update for the INS Kalman.

Test results of the proposed INS/Vision measurement fusion lead to the development of an efficient fusion approach to update the EKF with the position estimated from the image matching employing the residuals originating from the object space transformation parameters estimation as the error covariance matrix in the EKF implementation with inertial measurement for updating navigation for the UAV. The proposed algorithm led to improvement in the navigation solution with position updates from vision measurements compared to the navigation solution during GPS signal outage with 91.72% in the east position error and 85.56% in the north position error.

6.3 Future work

The thesis introduced an efficient navigation approach based on visual measurements taken from onboard camera mounted on the UAV. Some of the interesting research topics recommended for future work are as follows:

1. The work done in this thesis was based on simulated measurements. Following the previous conclusions, it is recommended to implement the proposed VBN system on real UAV with low cost sensors.
2. Based on GPU CUDA efficient implementation, generating DEM models using onboard sensors should be investigated. Such DEM models are needed for matching with pre-stored DEM models for the flying area which will help to extract the necessary information required for navigation where fewer landmarks are available.
3. Based on GPU CUDA efficient implementation, the concept of optical flow should be investigated for pose estimation of the UAV.

REFERENCES

Alkaabi, S. and F. Deravi (2005). "Block matching in rotated images." ELECTRONICS LETTERS- IEE **41**(4): 181.

Anderson, B. D. O. and J. B. Moore (1979). Optimal filtering. Englewood Cliffs, N.J., Prentice-Hall.

Anqi, X. and G. Dudek (2010). A vision-based boundary following framework for aerial vehicles. Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on.

Barron, J. L., D. J. Fleet, S. S. Beauchemin and T. A. Burkitt (1992). Performance of optical flow techniques. Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on.

Bay, H., A. Ess, T. Tuytelaars and L. Van Gool (2008). "Speeded-Up Robust Features (SURF)." Computer Vision and Image Understanding **110**(3): 346-359.

Bay, H., T. Tuytelaars and L. Van Gool (2006). SURF: Speeded up robust features. Computer Vision - Eccv 2006 , Pt 1, Proceedings. A. Leonardis, H. Bischof and A. Pinz. Berlin, Springer-Verlag Berlin. **3951:** 404-417.

Berg, A. C., T. L. Berg and J. Malik (2005). Shape matching and object recognition using low distortion correspondences. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.

Bhanu, B., B. Roberts and J. Ming (1990). Inertial navigation sensor integrated motion analysis for obstacle detection. Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on.

Bjorke, K. (2006). "Image processing on parallel GPU pixel units [6065-41]." PROCEEDINGS-SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING **6068**: 606515.

Borgefors, G. (1988). "Hierarchical chamfer matching: a parametric edge matching algorithm." Pattern Analysis and Machine Intelligence, IEEE Transactions on **10**(6): 849-865.

Brown, M. and D. Lowe (2002). "Invariant Features from Interest Point Groups."

Brown, R. G. and P. Y. C. Hwang (1992). Introduction to random signals and applied Kalman filtering. New York, J. Wiley.

Brown, R. G. and P. Y. C. Hwang (1997). Introduction to random signals and applied kalman filtering. New York; Toronto, J. Wiley.

Broyden, C. G. (1970). "The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations." IMA Journal of Applied Mathematics **6**(1): 76-90.

Conn, A. R., N. I. M. Gould and P. L. Toint (2000). Trust-region methods. Philadelphia, PA, Society for Industrial and Applied Mathematics.

Di Stefano, L., S. Mattoccia and F. Tombari (2005). "ZNCC-based template matching using bounded partial correlation." PATTERN RECOGNITION LETTERS **26**(14): 2129-2134.

Dong-Gyu, S., P. Rae-Hong, K. Rin-Chul, L. Sang Uk, et al. (2002). "Integrated position estimation using aerial image sequences." Pattern Analysis and Machine Intelligence, IEEE Transactions on **24**(1): 1-18.

El-Habiby, M. M., Y. Gao and M. G. Sideris (2009). "Comparison and Analysis of Non-Linear Least Squares Methods for 3-D Coordinates Transformation." Survey review. **41**(311): 26.

El-Sheimy (2007). Inertial Techniques and INS/DGPS Integration (lecture notes). ENGO 623.

El Oirrak, A., M. Daoudi and D. Aboutajdine (2002). "Estimation of general 2D affine motion using Fourier descriptors." Pattern recognition. **35**: 223-228.

Essannouni, F., R. O. H. Thami, D. Aboutajdine and A. Salam (2007). "Fast L4 template matching using frequency domain." ELECTRONICS LETTERS- IEE **43**(9): 507.

Eun-Hwan, S. and N. El-Sheimy (2004). "An unscented Kalman filter for in-motion alignment of low-cost IMUs." Position Location and Navigation Symposium PLANS 2004.

Fitch, A. J., A. Kadyrov, W. J. Christmas and J. Kittler (2005). "Fast robust correlation." Image Processing, IEEE Transactions on **14**(8): 1063-1073.

Fletcher, R. (2000). Practical methods of optimization. Chichester, Wiley.

Hagen, E. and E. Heyerdahl (1992). Navigation by optical flow. Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on.

Harris, C. B. (1989). Prototype for a land based Automatic Vehicle Location and Navigation System. Canada, University of Calgary (Canada). **M.Eng.:** 178-178 p.

Heiskanen, W. A. and H. Moritz (1967). Physical geodesy. San Francisco, W.H. Freeman.

Jalil, A., I. M. Qureshi, A. Manzar, R. A. Zahoor, et al. (2006). Rotation-Invariant Features for Texture Image Classification. Engineering of Intelligent Systems, 2006 IEEE International Conference on.

Jekeli, C. (2001). Inertial navigation systems with geodetic applications. Berlin; New York, Walter de Gruyter.

Johnson, N. L. and S. Kotz (1997). Leading personalities in statistical sciences : from the seventeenth century to the present. New York, Wiley.

Krakiwsky, E. J. (1995). IVHS navigation systems database. Calgary, University of Calgary, Dept. of Geomatics Engineering.

Krakiwsky, E. J. and D. Wells (1971). Coordinate systems in geodesy. Fredericton, Dept. of Surveying Engineering, University of New Brunswick.

Kruis, M. (2010). Human pose recognition using neural networks, synthetic models, and modern features. Elecrtical Engineering. Stillwater, OK, Oklahoma State University. **Master Of Science** 64.

Lagarias, J. C., J. A. Reeds, M. H. Wright and P. E. Wright (1998). "Convergence properties of the Nelder-Mead simplex method in low dimensions." Siam Journal on Optimization **9**(1): 112-147.

Lerner, R., E. Rivlin and P. H. Rotstein (2004). Error analysis for a navigation algorithm based on optical-flow and a digital terrain map. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on.

Lowe, D. G. (1991). "Fitting parameterized three-dimensional models to images." Pattern Analysis and Machine Intelligence, IEEE Transactions on **13**(5): 441-450.

Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints." International Journal of Computer Vision **60**(2): 91-110.

Lu, C. P., G. D. Hager and E. Mjolsness (2000). "Fast and globally convergent pose estimation from video images." Ieee Transactions on Pattern Analysis and Machine Intelligence **22**(6): 610-622.

Luenberger, D. G. (1984). Linear and nonlinear programming. Reading, Mass., Addison-Wesley.

Makarov, A. M. M. (2006). "Binary shape coding using finite automata." IEE proceedings. Vision, image, and signal processing. **153**(5): 695.

Marquardt, D. W. (1963). "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." Journal of the Society for Industrial and Applied Mathematics **11**(2): 431-441.

Mattoccia, S., F. Tombari, L. D. Stefano and M. Pignoloni (2007). Efficient and optimal block matching for motion estimation. Proceedings of the 14th International Conference on Image Analysis and Processing, IEEE Computer Society**:** 705-710.

McCarthy, C. and N. Barnes (2004). Performance of optical flow techniques for indoor navigation with a mobile robot. Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on.

Mikolajczyk, K. and C. Schmid (2001). Indexing based on scale invariant interest points. Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on.

Morita, S. (1997). Generating stable structure using Scale-space analysis with non-uniform Gaussian kernels

Scale-Space Theory in Computer Vision. B. ter Haar Romeny, L. Florack, J. Koenderink and M. Viergever, Springer Berlin / Heidelberg. **1252:** 89-100.

Nelder, J. A. and R. Mead (1965). "A SIMPLEX-METHOD FOR FUNCTION MINIMIZATION." Computer Journal **7**(4): 308-313.

Nilsson, J. (2005). Visual Landmark Selection and Recognition for Autonomous Unmanned Aerial Vehicle Navigation. Department of Numerical Analysis and Computer Science. Stockholm, Sweden, Royal Institute of Technology. **Master's Degree Project:** 46.

Nocedal, J. and S. J. Wright (1999). Numerical optimization. New York, Springer.

Olson, C. F., L. H. Matthies, H. Schoppers and M. W. Maimone (2000). Robust stereo ego-motion for long distance navigation. Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on.

Pachter, M. and A. Porter (2004). Bearings-only measurements for INS aiding: the three-dimensional case. American Control Conference, 2004. Proceedings of the 2004.

Pharr, M. and R. Fernando (2005). GPU gems 2 : programming techniques for high-performance graphics and general-purpose computation. Upper Saddle River, NJ, Addison-Wesley.

Powell, M. J. D. and H. Atomic Energy Research Establishment (1968). A FORTRAN SUBROUTINE FOR SOLVING SYSTEMS OF NONLINEAR ALGEBRAIC EQUATIONS. United Kingdom.

Raquet, J. F., Giebner, M. (2003). Navigation Using Optical Measurements of Objects at Unknown Locations,. Proceedings of the 59th Annual Meeting of The Institute of Navigation and CIGTF 22nd Guidance Test Symposium, Albuquerque, NM,.

Roberts, B. and B. Bhanu (1992). "Inertial navigation sensor integrated motion analysis for autonomous vehicle navigation." Journal of Robotic Systems **9**(6): 817-842.

Roumeliotis, S. I., A. E. Johnson and J. F. Montgomery (2002). Augmenting inertial navigation with image-based motion estimation. Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on.

Sheta , B., M. Elhabiby and N. El-Sheimy (2012a). "Assessment of nonlinear optimization and Speeded Up Robust Features (SURF) algorithm for estimating object space transformation parameters for UAV pose estimation." Geomatica **66**(4).

Sheta , B., M. Elhabiby and N. El-Sheimy (2012b). "Assessments of different Speeded Up Robust Features (SURF) algorithm resolution for pose estimation of UAV." International Journal Of Computer science and Engineering Survey.

Sheta , B., M. Elhabiby and N. El-Sheimy (2012c). Assessments of nonlinear least squares methods for UAV Vision Based Navigation (VBN). ASPRS Annual Meeting Sacramento, California, USA. .

Sheta , B., M. Elhabiby and N. El-Sheimy (2012d). Comparison and analysis of nonlinear least squares methods for Vision Based Navigation (VBN) algorithms. XXII ISPRS Congress , , Melbourne, Australia.

Sheta , B., A. Elsharkawy, M. Elhabiby and N. El-Sheimy (2012e). Parallel computation for Remote Sensing and VBN systems processing – CUDA approach. CGU Annual Scientific Meeting, . Banff, Canada.

Sim, D. G., S. Y. Jeong, R. H. Park, R. C. Kim, et al. (1996). Navigation parameter estimation from sequential aerial images. Image Processing, 1996. Proceedings., International Conference on.

Stepanyan, V. (2006). "Vision based guidance and flight control in problems of aerial tracking." from http://scholar.lib.vt.edu/theses/available/etd-08142006-151453.

Teunissen, P. J. G. (1990). "Nonlinear least squares." Manuscripta Geodaetica **15**(Copyright 1990, IEE): 137-150.

Titterton, D. H. and J. L. Weston (1997). Strapdown inertial navigation technology. Stevenage, Herts., Peregrinus.

Titterton, D. H., J. L. Weston, E. Institution of Electrical, A. American Institute of, et al. (2004). "Strapdown inertial navigation technology." from http://www.knovel.com/knovel2/Toc.jsp?BookID=1241.

Venkataraman, P. (2002). Applied optimization with MATLAB programming. New York, Wiley.

Veth, M. J. (2006). Fusion of imaging and inertial sensors for navigation.

Viola, P. and M. Jones (2001). Rapid object detection using a boosted cascade of simple features. 2001 Ieee Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1, Proceedings. A. Jacobs and T. Baldwin. Los Alamitos, Ieee Computer Soc**:** 511-518.

Wolf, P. R. and B. A. Dewitt (2000). Elements of photogrammetry : with applications in GIS. Boston, McGraw-Hill.

Www.Spirent.com. "Integrated GPS/Inertial Test."

APPENDIX A

Results of the developed modified SURF algorithm for the proposed VBN system are shown below. Results show the developed modified SURF algorithm for descriptor length 128 with scale variations 0.2 to 1 with rotation angle 15°. Moreover, results for the developed modified SURF algorithm are presented for descriptor lengths 36 and 64 with scale variation 0.2 to 1 with rotation angle 45° and 9×9 sample points in each sub-region.

Results clearly show the robustness of the developed modified SURF algorithm against scale and rotation variation which is a key element for the developed VBN system.

Figure A 1: Descriptor length 128 with scale variation = 0.2 and rotation = 15 and number of sample points 5x5

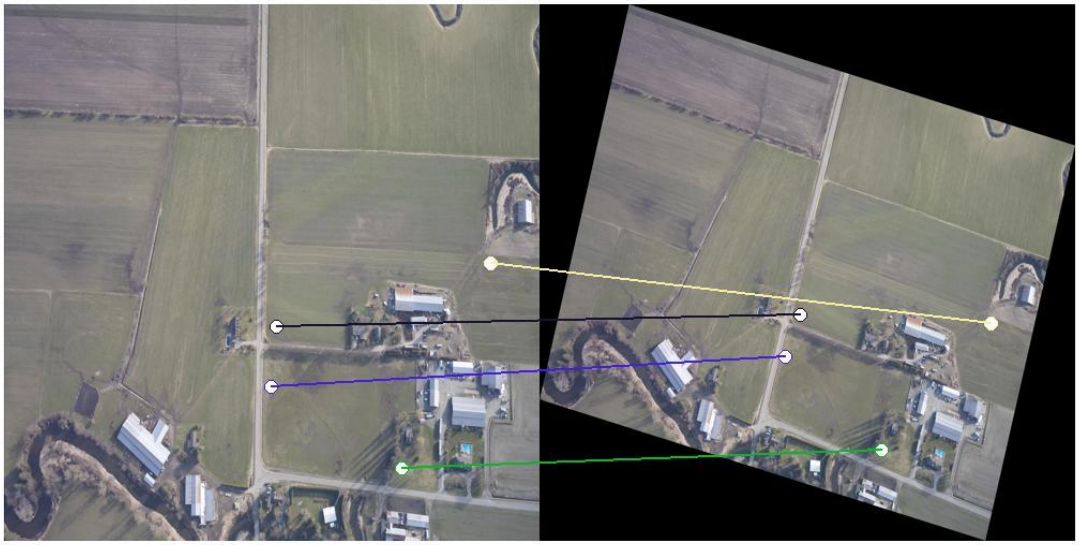


Figure A 2: Descriptor length 128 with scale variation = 0.4 and rotation = 15 and number of sample points 5x5

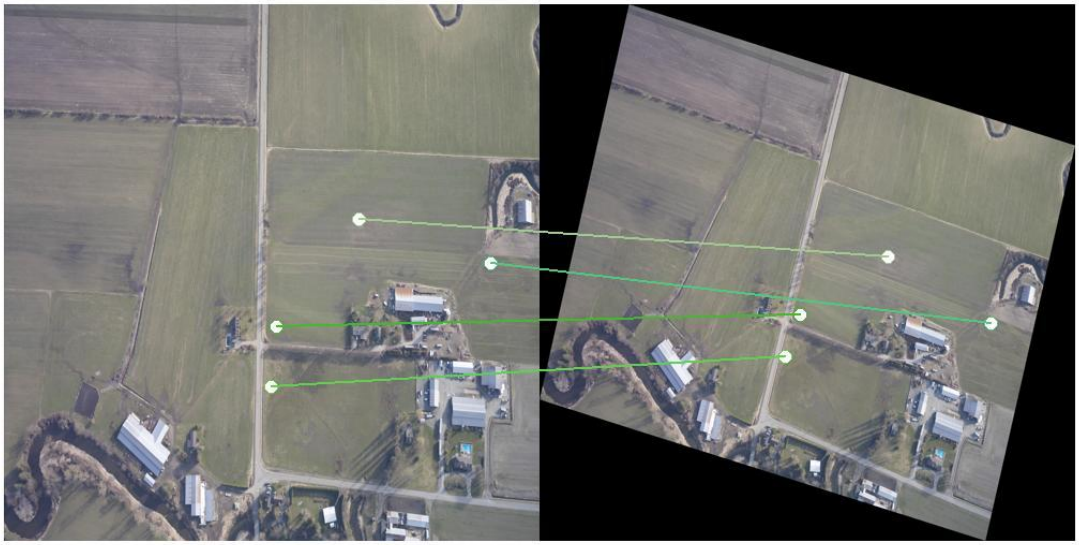


Figure A 3: Descriptor length 128 with scale variation = 0.6 and rotation = 15 and number of sample points 5x5

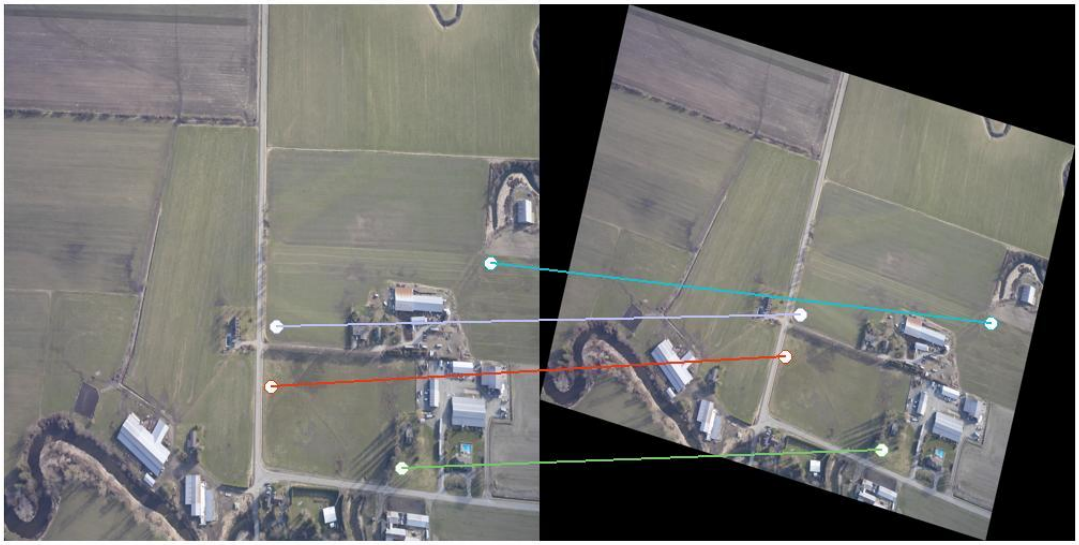


Figure A 4: Descriptor length 128 with scale variation = 0.8 and rotation = 15 and number of sample points 5x5

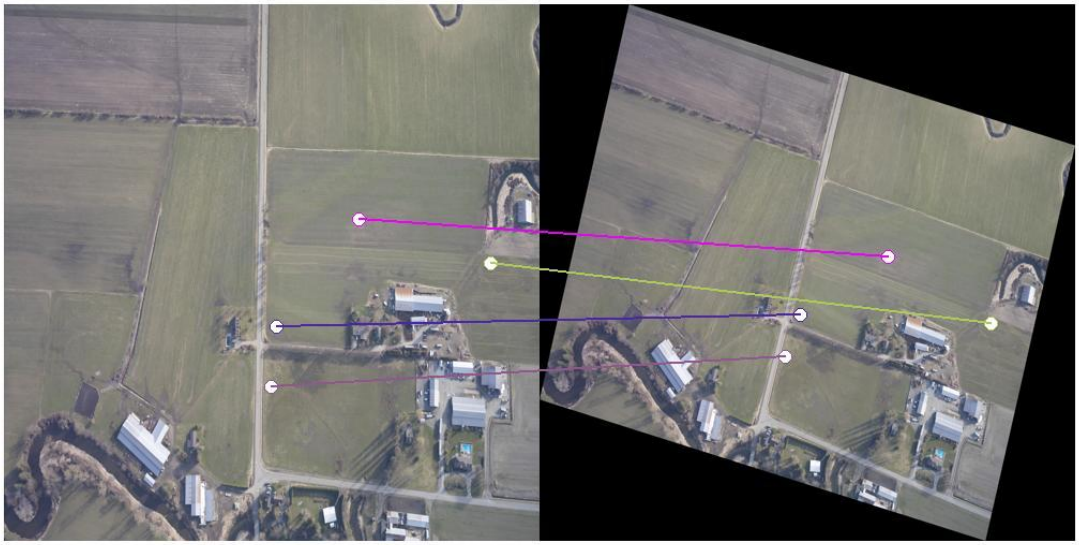


Figure A 5: Descriptor length 128 with scale variation = 0.8 and rotation = 15 and number of sample points 5x5

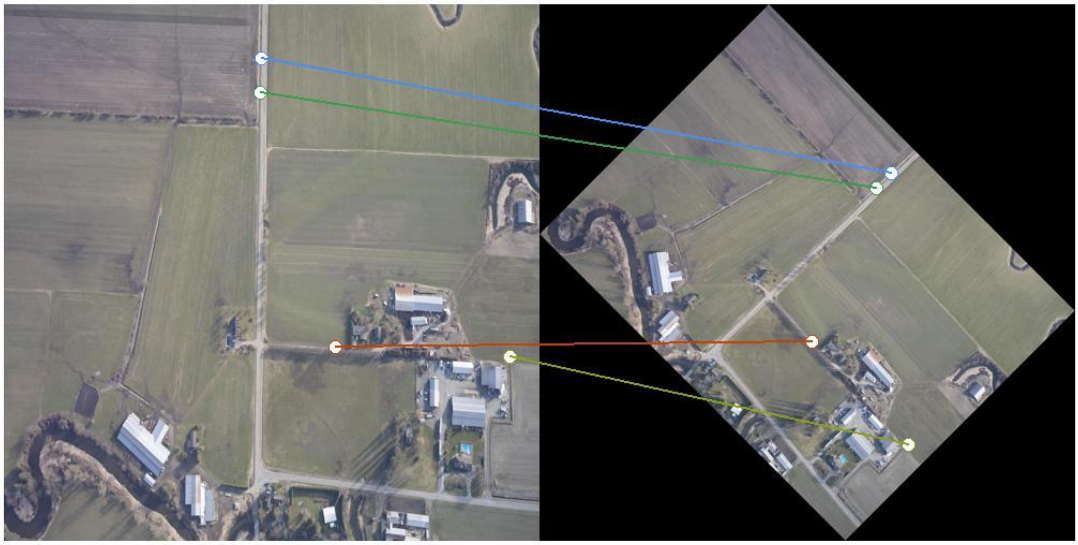


Figure A 6: Descriptor length 64 with scale variation = 0.2 and rotation = 45 and number of sample points 5x5

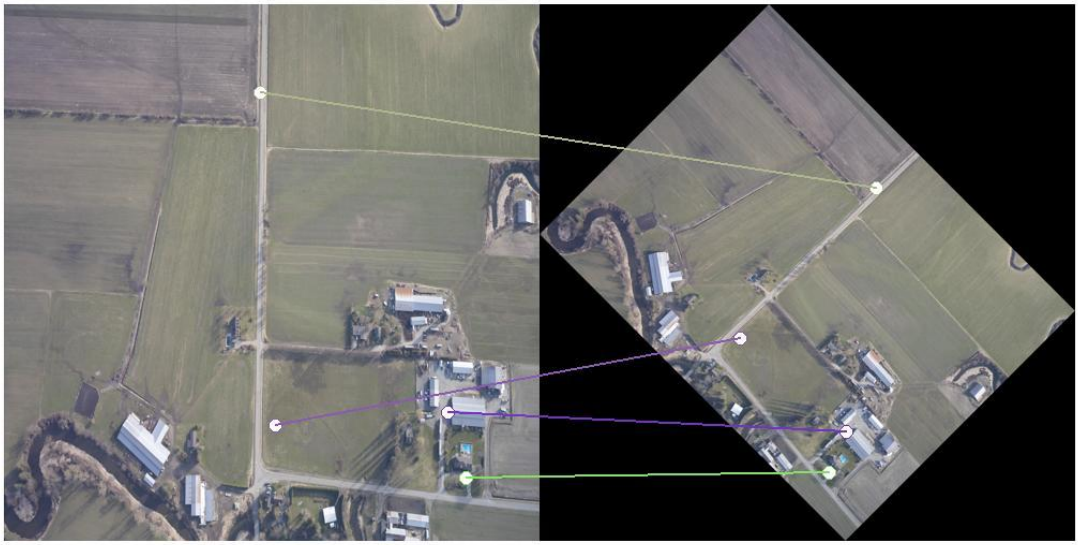


Figure A 7: Descriptor length 64 with scale variation = 0.4 and rotation = 45 and number of sample points 5x5

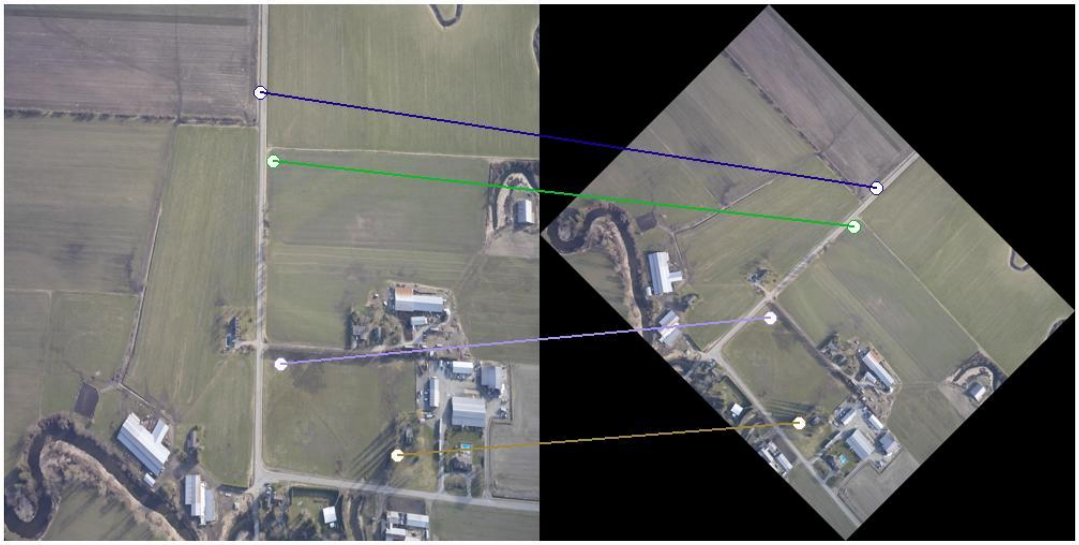


Figure A 8: Descriptor length 64 with scale variation = 0.6 and rotation = 45 and number of sample points 5x5

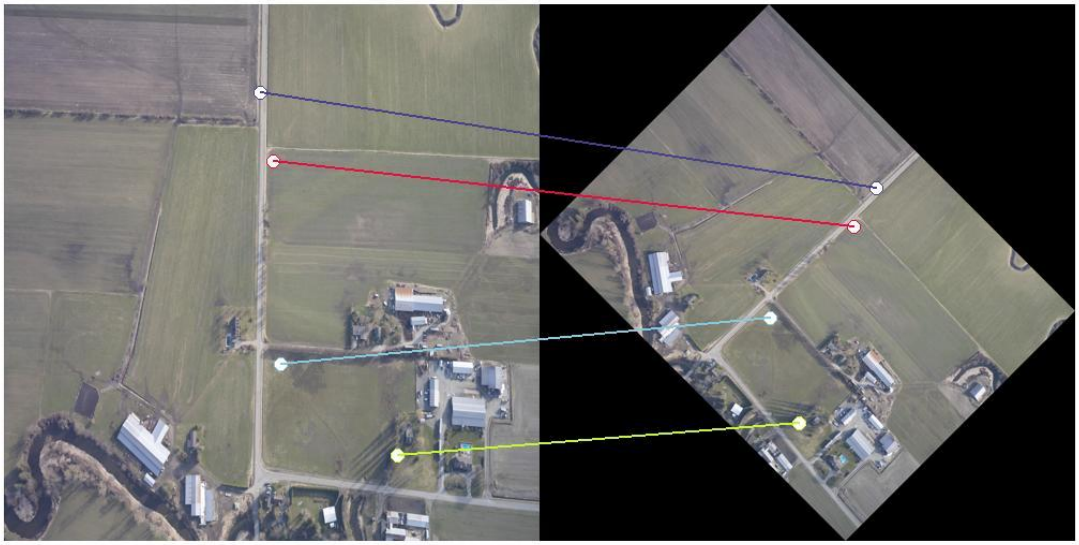


Figure A 9: Descriptor length 64 with scale variation = 0.8 and rotation = 45 and number of sample points 5x5

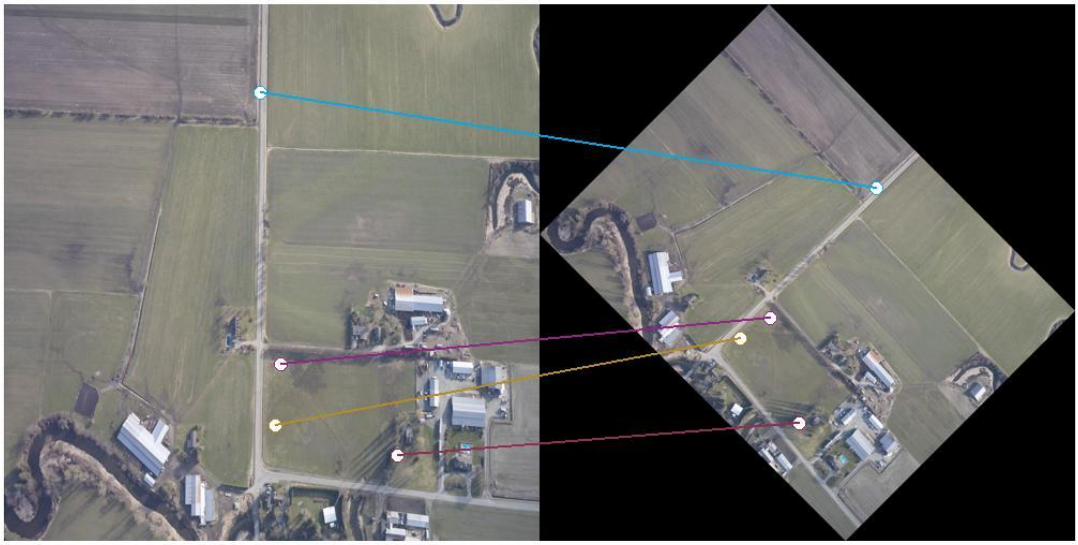


Figure A 10: Descriptor length 64 with scale variation = 1 and rotation = 45 and number of sample points 5x5

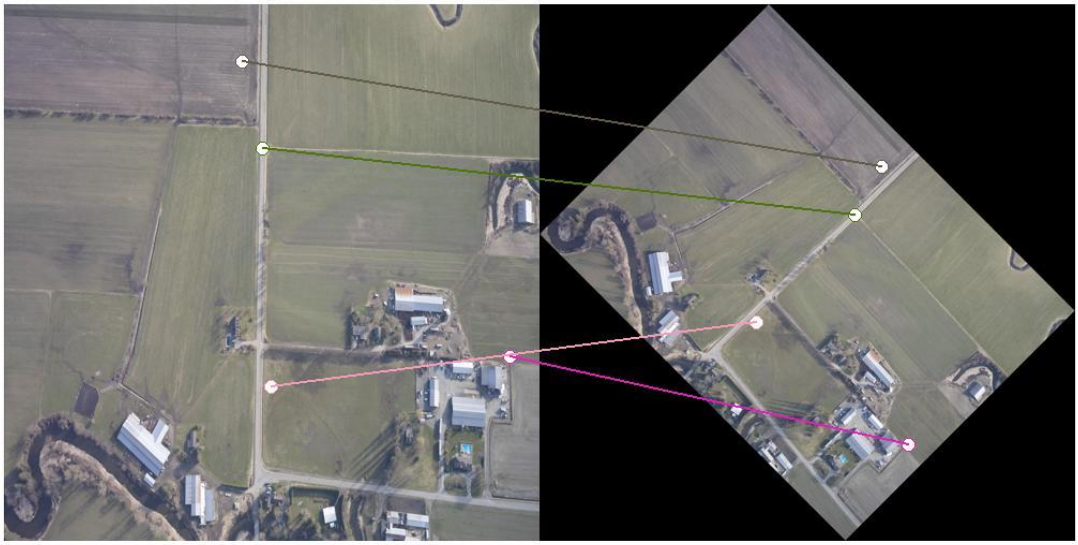


Figure A 11: Descriptor length 36 with scale variation = 0.2 and rotation = 45 and number of sample points 5x5

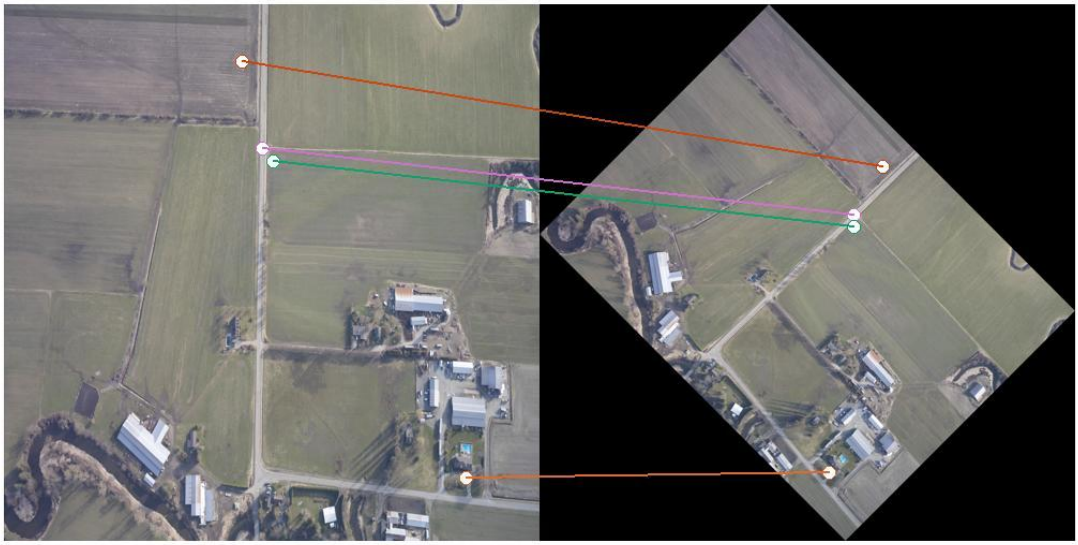


Figure A 12: Descriptor length 36 with scale variation = 0.4 and rotation = 45 and number of sample points 5x5

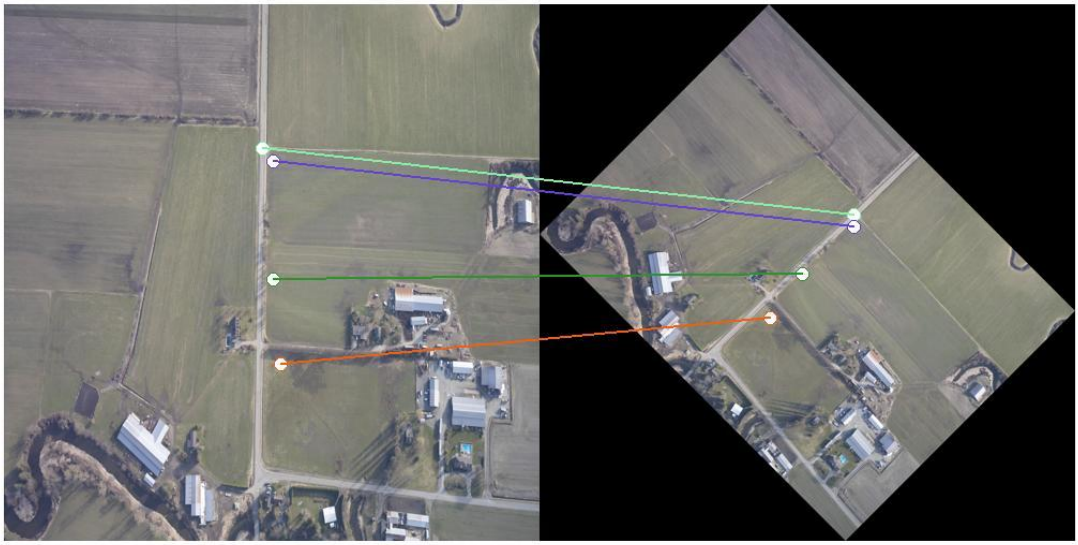


Figure A 13: Descriptor length 36 with scale variation = 0.6 and rotation = 45 and number of sample points 5x5

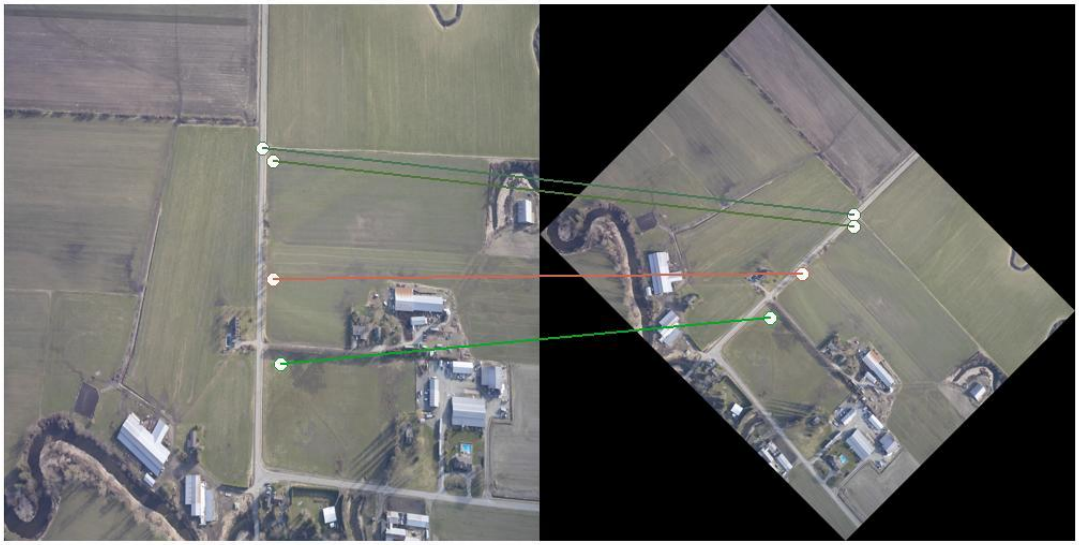


Figure A 14: Descriptor length 36 with scale variation = 0.8 and rotation = 45 and number of sample points 5x5

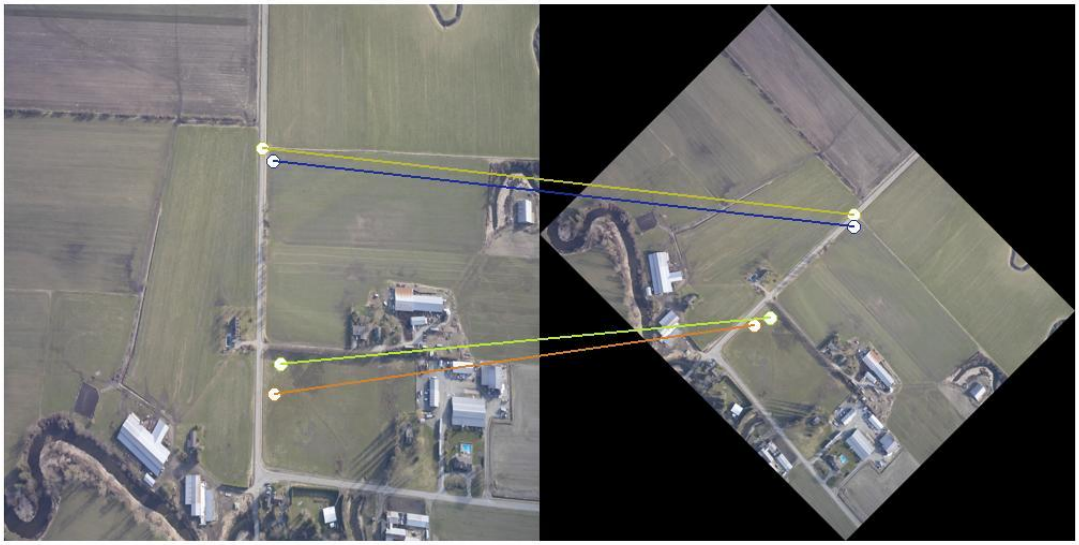


Figure A 15: Descriptor length 36 with scale variation = 1 and rotation = 45 and number of sample points 5x5

Figure A 16: Descriptor length 64 with scale variation = 0.2 and rotation =15 and number of sample points 9x9

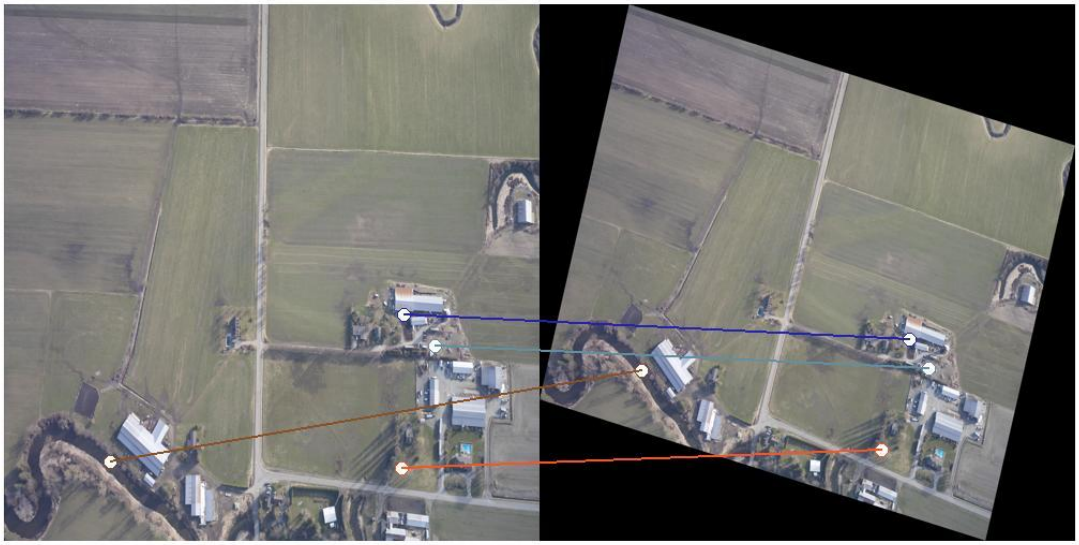


Figure A 17: Descriptor length 64 with scale variation = 0.4 and rotation =15 and number of sample points 9x9

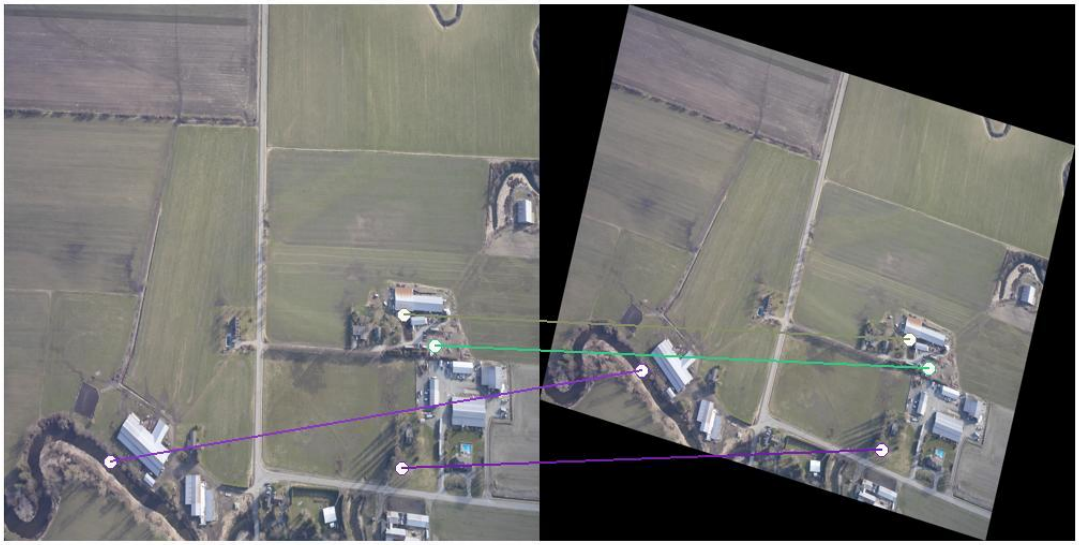


Figure A 18: Descriptor length 64 with scale variation = 0.6 and rotation =15 and number of sample points 9x9

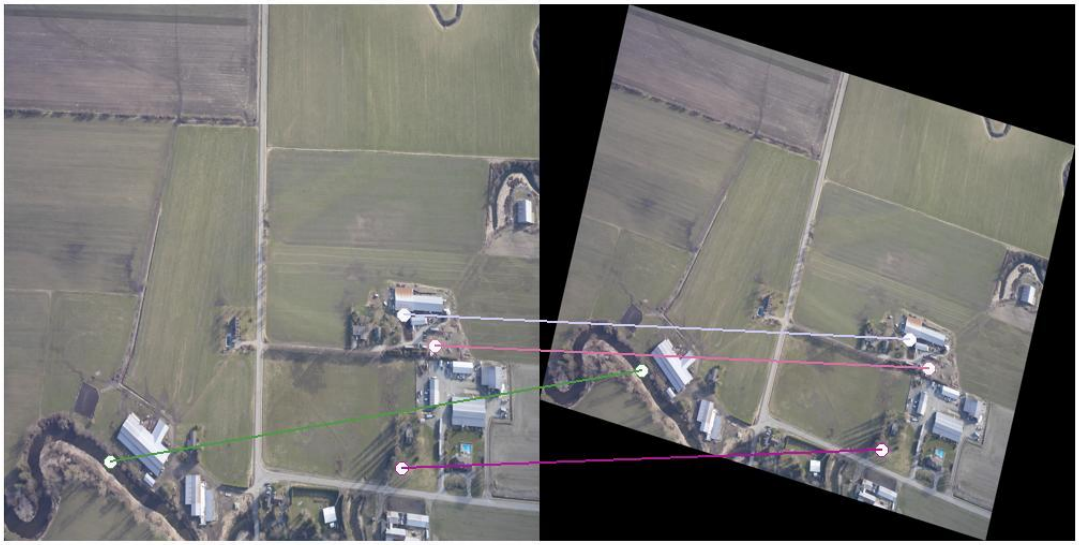


Figure A 19: Descriptor length 64 with scale variation = 0.8 and rotation =15 and number of sample points 9x9

Figure A 20: Descriptor length 64 with scale variation = 1 and rotation =15 and number of sample points 9x9

Figure A 21: Descriptor length 36 with scale variation = 0.2 and rotation =15 and number of sample points 9x9

Figure A 22: Descriptor length 36 with scale variation = 0.4 and rotation =15 and number of sample points 9x9

Figure A 23: Descriptor length 36 with scale variation = 0.6 and rotation =15 and number of sample points 9x9

Figure A 24: Descriptor length 36 with scale variation = 0.8 and rotation =15 and number of sample points 9x9

Figure A 25: Descriptor length 36 with scale variation = 0.8 and rotation =15 and number of sample points 9x9

APPENDIX B

Results for the developed nonlinear optimization algorithms are presented below. Convergence of the developed nonlinear optimization algorithms is tested with general tilted photograph with different values for angle φ varying from φ = 25° to 45° with step 5°.

Results clearly show that the Nelder-Mead simplex direct search method failed to estimate the object space transformation parameters because this method depends on the objective function only without any derivative information. However, the object space transformation parameters were successfully estimated only by Quasi-Newton line search method due to depending on the Hessian matrix of the Collinearity equations.

Table B 1: Performance of optimization methods with initial φ (deg) = 25

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.054	0.057	0.036	0.021	0.007
$\delta\varphi$ (rad)	0.039	0.417	0.722	0.049	0.367
δK (rad)	0.028	0.297	0.636	0.021	0.081
δX_L (m)	32.591	269.341	299.053	17.933	317.632
δY_L (m)	8.382	110.221	355.848	6.058	12.676
δZ_L (m)	6.332	207.172	526.124	1.417	96.331
No. of Iteration	5	6	41	1	145

Table B 2: Performance of optimization methods with initial φ (deg) = 30

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.061	0.081	0.051	0.005	0.007
$\delta\varphi$ (rad)	0.049	0.605	0.701	0.049	0.372
δK (rad)	0.031	0.496	0.611	0.061	0.032
δX_L (m)	38.772	327.116	303.414	17.933	303.152
δY_L (m)	8.963	252.292	335.329	6.058	72.918
δZ_L (m)	9.808	396.564	502.321	1.417	9.056
No. of Iteration	7	7	35	2	152

Table B 3: Performance of optimization methods with initial φ (deg) = 35

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.054	0.08	0.064	0.008	0.007
$\delta\varphi$ (rad)	0.092	0.39	0.681	0.048	0.409
δK (rad)	0.06	0.285	0.583	0.099	0.12
δX_L (m)	67.948	272.93	307.632	17.933	313.834
δY_L (m)	13.081	141.619	317.527	6.058	55.491

δZ_L (m)	27.114	204.52	478.298	1.417	43.651
No. of Iteration	8	7	31	3	145

Table B 4: Performance of optimization methods with initial φ (deg) = 40

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.061	0.093	0.068	0.022	0.007
$\delta\varphi$(rad)	0.056	0.579	0.673	0.048	0.521
δK (rad)	0.045	0.468	0.574	0.136	0.231
δX_L (m)	41.978	332.266	308.877	17.933	386.565
δY_L (m)	10.81	266.633	312.289	6.058	125.136
δZ_L (m)	11.415	388.217	470.728	1.417	112.495
No. of Iteration	7	7	29	3	163

Table B 5: Performance of optimization methods with initial φ (deg) = 45

Object Space parameters errors	Trust Region Dogleg	Trust Region	Levenberg-Marquardt	Quasi-Newton line search	Nelder-Mead simplex direct search
$\delta\omega$ (rad)	0.062	0.093	0.102	0.033	0.007

$\delta\varphi$ (rad)	0.054	0.589	0.591	0.047	0.418
δK (rad)	0.047	0.487	0.471	0.167	0.268
δX_L (m)	42.567	323.951	316.447	17.932	310.099
δY_L (m)	12.072	281.309	263.707	6.058	160.33
δZ_L (m)	11.593	410.709	381.079	1.417	17.023
No. of Iteration	8	8	23	3	170