Cranfield University

Abdelkrim Nemra

Robust Airborne 3D Visual Simultaneous Localisation And Mapping

Cranfield Defence and Security

PhD

Cranfield University

Cranfield Defence and Security

Department of Informatics and Systems Engineering

PhD

2010

Abdelkrim Nemra

Robust Airborne 3D Visual Simultaneous Localisation And Mapping

Supervisor: Dr Nabil Aouf

16th September 2010

Abstract

The aim of this thesis is to present robust solutions to technical problems of airborne three-dimensional (3D) Visual Simultaneous Localisation And Mapping (VSLAM). These solutions are developed based on a stereovision system available onboard Unmanned Aerial Vehicles (UAVs). The proposed airborne VSLAM enables unmanned aerial vehicles to construct a reliable map of an unknown environment and localise themselves within this map without any user intervention.

Current research challenges related to Airborne VSLAM include the visual processing through invariant feature detectors/descriptors, efficient mapping of large environments and cooperative navigation and mapping of complex environments. Most of these challenges require scalable representations, robust data association algorithms, consistent estimation techniques, and fusion of different sensor modalities. To deal with these challenges, seven Chapters are presented in this thesis as follows: Chapter 1 introduces UAVs, definitions, current challenges and different applications. Next, in Chapter 2 we present the main sensors used by UAVs during navigation. Chapter 3 presents an important task for autonomous navigation which is UAV localisation. In this chapter, some robust and optimal approaches for data fusion are proposed with performance analysis. After that, UAV map building is presented in Chapter 4. This latter is divided into three parts. In the first part, a new imaging alternative technique is proposed to extract and match a suitable number of invariant features. The second part presents an image mosaicing algorithm followed by a super-resolution approach. In the third part, we propose a new feature detector and descriptor that is fast, robust and detect suitable number of features to solve the VSLAM problem. A complete Airborne Visual Simultaneous Localisation and Mapping (VSLAM) solution based on a stereovision system is presented in Chapter (5). Robust data association filters with consistency and observability analysis are presented in this chapter as well. The proposed algorithm is validated with loop closing detection and map management using experimental data. The airborne VSLAM is extended then to the multiple UAVs case in Chapter (6). This chapter presents two architectures of cooperation: a Centralised and a Decentralised. The former provides optimal precision in terms of UAV positions and

constructed map while the latter is more suitable for real time and embedded system applications. Finally, conclusions and future works are presented in Chapter (7).

To my parents, Abdelkader and Fatima

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr Nabil Aouf for his continuous guidance, support and patience as well as giving me the chance to undertake this research in the first place. I would like also to express a special thanks to General Amara Mohamed Najib Head of Military Polytechnic School of Algeria for giving me a chance to do PhD in the Defence Academy of United Kingdom. I am also grateful to my committee thesis members, Dr Mark Richardson and Dr Peter Yuen for their supportive approach in discussing and reviewing my progress in addition to their invaluable comments and suggestions.

I wish to extend my special thanks to my parents, my wife, brothers and my sisters, who all have given me their endless love and support.

I would like also to acknowledge and thank all the staff at the Military Polytechnic School, Algeria especially the brilliant people of control unit and the robotics group and the staff at the Cranfield University/Defence Academy, UK with whom I have had constant communication for the past three years.

Many thanks are reserved to all of my friends and student colleagues at the Defence Academy / Cranfield Defence and Security, Shrivenham. The journey has been remarkable and every one of them has made it a better experience. I have had the pleasure of being in a company with some of the kindest people from all walks of life.

Finally, I would like to thank all people who have contributed to the achievement of this work

Shrivenham, 16th September 2010 Abdelkrim Nemra

Contents

List of Figures

List of Tables

Nomenclature

Roman symbols

ax, ay, az	IMU acceleration
p, q, r	IMU angular rates
X, Y, Z	UAV position in navigation frame
U, V, W	UAV velocity in body frame
C_{bn}	Direct cosine transform matrix that rotates a vector from body frame to the navigation frame
P_k	Variance covariance matrix
$\hat{x}_{k/k}$	Estimated state at time step k
$\hat{x}_{k/k-1}$	Predicted state
w_k	Process noise
v_k	Observation noise
Q_k	Process noise covariance matrix
R_k	Observation noise covariance matrix
y_k	Observation
$f(\cdot, \cdot, \cdot)$	Continuous Process model
$h(\cdot, \cdot, \cdot)$	Continuous Observation model
K_k	Kalman Gain
c_w	Process noise scale factor in $NH\infty$ filter
c_v	Observation noise scale factor in $NH\infty$ filter
$V(\hat{x})$	Lyapunov function
$c(x, y)$	Harris auto-correlation function
$C(x, y)$	Second moment matrix
E	Energy of the image
k	Scale factor for image pyramid

I	Original image
E_s	Energy threshold
f	Focal length
b	Baseline
db	Daubechies
(u_l, v_l)	Feature coordinate in left image
(u_r, v_r)	Feature coordinate in right image
H	Homography matrix
A_k	Approximation image from wavelet decomposition
H_k, V_k, D_k	Horizontal, Vertical and Diagonal details respectively from wavelet decomposition.
$f_i = [u_i, v_i, k_i]$	Feature coordinates in scale and space
N_f	Norm of the dominant orientation
ms	Window of description
bin	Orientation sampling
I_{c1}, I_{c2}	Intrinsic parameters of camera 1 and 2 respectively
k_v	Horizontal scale factor
k_u	Vertical scale factor
(u_0, v_0)	Coordinate of Optical Center
C_b^n	Rotation matrix from body to navigation frame
C_s^b	Rotation matrix from IMU to body frame
C_{c1}^s (C_{c2}^s)	Rotation matrix from the right (left) camera to the IMU frame
O	Observability matrix
co	Consistency parameter
Rco	Robust consistency parameter
E_{th}	Energy threshold for ASIFT
$HD_{\max}$	Maximal Horizontal Disparity

$HV_{\max}$	Maximal Vertical Disparity

Greek symbols

$\phi,\ \theta,\ \psi$	UAV orientation in navigation frame
∇	Jacobean
Δ_i	High order term in Taylor development
γ	H infinity bound
δ_i	Bounds of high order terms in Taylor development
$\lambda 1, \lambda 2$	Eigen values of Second moment matrix
σ	Image scale
$\alpha_r, \alpha_g, \alpha_b$	Contrast factors for RGB channels respectively
$\beta_r, \beta_g, \beta_b$	Luminosity factors for RGB channels respectively
$\Delta\theta$	Orientation step
Θ_f	Dominant orientation

Acronyms

AI	Artificial Intelligence
ACI	Adaptive Covariance Intersection
ASIFT	Adaptive Scale Invariant Feature Transform
CI	Covariance intersection
CML	Concurrently Localisation and Mapping
C-VSLAM	Cooperative Visual Simultaneous Localisation And Mapping
CC-VSLAM	Centralised Cooperative Visual SLAM
DC-VSLAM	Decentralised Cooperative Visual SLAM
DE	Distributed estimation
DistRatio	Distance ration
DoD	Department of Defence
DoF	Degree of Freedom
DoG	Difference of Gauss
DOP	Dilution of Precision
DWT	Discrete Wavelet Transform

EKF	Extended Kalman Filter
FCI	Fast Covariance Intersection
HD	Horizontal Disparity
HR	High resolution
IMO	Inverse Model of Observation
IMU	Inertial Measurement Unit
INS	Inertial Measurement System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
KF	Kalman Filter
KLT	Kanade Lucas Tomasi
LoG	Laplacian of Gauss
LR	Low resolution
MMSE	Minimum Mean Square Error
MS	Matching score
NED	Navigation East Down
NEF	Number of extracted features
$NH\infty$	Nonlinear H ∞
NGM	Number of good matching
PF	Particle Filter
PTZ	Pan Tilt Zoom
ROI	Region of Interest
RPV	Remotely Piloted Vehicle
SIFT	Scale Invariant Feature Transform
SURF	Speed Up Robust Features
SWIF	Scale Wavelet Invariant Transform
SDC	State Dependent Coefficient
SDRE	State Dependent Ricatti Equation
SLAM	Simultaneous Localisation And Mapping
SM	Stochastic Map
SVD	Singular Value Decomposition

S-VSLAM	Single Visual Simultaneous Localisation And Mapping
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
UT	Unscented Transform
VNC	Variation of Number of Correspondences
VSLAM	Visual Simultaneous Localisation And Mapping
VD	Vertical Disparity

Chapter 1

Introduction

CHAPTER 1

Introduction

Unmanned Aerial Vehicles (UAVs) are rapidly becoming a strategic asset of today's military forces and civilian airspace community. They are encountered in an increasing number of different applications like: surveillance, reconnaissance, communication relay, target designation and payload delivery. The term 'Unmanned Aerial Vehicle' (UAV) encompasses a wide variety of robotic aircraft that vary in size, shape, flight characteristics and level of operational autonomy. In recent years, the development and production of UAVs has undergone an explosion in the military arenas.

UAVs in Figure (1.1) have been referred under different names: RPVs (remotely piloted vehicles), drones, robot planes and pilotless aircraft are a few such names. UAVs are defined by the American Department of Defence (DOD) as powered aerial vehicles that do not carry a human operator, use aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or non lethal payload. Ballistic or semi ballistic vehicles, cruise missiles and artillery projectiles are not considered UAVs by the DOD definition. UAVs are either described as a single air vehicle (with associated surveillance sensors), or a UAV system, which usually consists of three to six air vehicles, a ground control station, and support equipments.

Figure.1. 1 Examples of UAVs

Today challenge for aerial vehicles is in how to achieving their full decision autonomy. Accuracy of these vehicles is then introduced as the necessary performance for safely completion mission. Increased accuracy can be achieved by accurate localisation within an accurate map. This leads to the problem of Simultaneous Localisation And Mapping (SLAM).

The problem of Simultaneous Localisation And Mapping for an autonomous vehicle is stated as follows: starting from an initial position, a UAV travels through a sequence of positions and obtains a set of sensor measurements at each position. The goal for the vehicle is to process the sensor data to produce an estimate of its position while concurrently building a map of its environment.

During the last decade, SLAM has been investigated as a significant research problem that was looked at often for simple 2D scenarios. Open SLAM challenges include efficient mapping of large environments, modelling complex and dynamic environments, multi-Airborne vehicle SLAM, full 3D SLAM and Cooperative SLAM. Most of these challenges will require scalable representations, robust data association algorithms, consistent estimation techniques, and different sensor modalities. In particular, solving SLAM with vision sensors is of crucial importance in order to address many real life applications.

The objective of this work is to assess current state-of the-art in SLAM, to discuss and to compare different approaches presenting recent advances, then proposing alternatives solutions to the SLAM problem, where the proposed algorithms are more robust, stable and adapted to real time applications. The work will cover a wide variety of classical and emerging imaging algorithms to detect distinctive, invariant and stable landmarks to construct a reliable map of the UAV environment. One important aspect of the SLAM problem which is investigated in this work is the data fusion techniques. Different approaches are implemented and compared, beginning by the popular Extended Kalman Filter, then the optimal Nonlinear State Dependant Riccati Equation (SDRE) and finally the robust nonlinear H∞ filter. The latter doesn’t make any assumption on noise characteristics; furthermore it is robust face error modelling which is crucial for real applications.

1.1 PhD Challenges

Autonomous vehicles must operate in their environment without any (or minimum) human intervention and meet rigorous requirements associated with any airborne platforms. Autonomy of these vehicles requires the development of navigation and guidance algorithms for self localisation and based on map environment building.

The increase of UAV autonomy can be achieved by developing tools able to provide an accurate automatic localisation in an accurate environment map. SLAM problem looks at the ability of an autonomous vehicle, starting in a partially known or unknown environment, to incrementally build an environment map and simultaneously localise itself within this map. New challenges in SLAM are delivering methods enabling large-scale implementations in increasingly unstructured environments and especially in situations where GPS-like solutions are unavailable or unreliable such as urban canyons and urban environments.

My PhD research study deals with integrated sensory data fusion and visual guidance for airborne systems. The main research areas focus on the following:

- Airborne localisation (data fusion filters and data association techniques).
- Map building (feature extraction and matching algorithms).
- Images registration and mosaicing.
- SLAM observability and consistency analysis.
- Robust airborne Visual SLAM.
- Cooperative VSLAM.

In principle, implementing SLAM in three dimensions (3D) is an extension of the two-dimensional (2-D) case. However 3D SLAM involves significant added complexity due to the more general vehicle motion model and most importantly increased sensing and feature modelling complexity. The challenge now is to demonstrate VSLAM solutions for a UAV navigating in large scale natural environment without recourse to global positioning system (GPS).

1.2 SLAM Problem State of Art

The objective of this work is to both develop and demonstrate autonomous localisation and mapping algorithms for airborne platforms. Autonomous localisation and mapping is the process of determining a platform's position, velocity and attitude information, and the map where the platform is navigating. These tasks should be completed without the use of any priori information external to the platform. Only for what the platform senses about the environment is used. This objective can be achieved by the implementation of a Simultaneous Localisation and Mapping (SLAM) algorithm.

The genesis of the probabilistic SLAM problem occurred in the IEEE Robotics and Automation Conference held in San Francisco, California in 1986. At that time, probabilistic methods were only just beginning to be introduced into both robotics and artificial intelligence (AI) problems. A number of researchers had been looking at applying estimation-theoretic methods to mapping and localisation problems. These included Peter Cheeseman, Jim Crowley and Hugh Durrant-Whyte. Over the course of the conference, many paper table cloths and napkins were filled with long discussions about consistent mapping. Along the way, Raja Chatila, Oliver Faugeras, Randal Smith and others also made useful contributions to the conversation. The result of these conversations was to recognise that consistent probabilistic mapping was a fundamental problem in robotics with major conceptual and computational issues that needed to be addressed. Over the next few years, a number of key papers were produced. Works by Smith and Cheesman [1] and Durrant-Whyte [2] established a statistical basis for describing relationships between landmarks and geometric uncertainty. A key element of these works was to show that there must be a high degree of correlation between estimates of the location of different landmarks in a map and that, indeed, these correlations would grow with successive observations.

At the same time Ayache and Faugeras [3] were undertaking early work in visual navigation, Crowley [4], Chatila and Laumond [5] were working on sonar-based navigation for mobile robots using Kalman filter type algorithms. These two strands of research had much in common and resulted soon after in the paper of Smith and al [6]. This paper showed that as a mobile robot moves through an unknown environment

based on relative observations of landmarks. The estimates of these landmarks are all necessarily correlated with each other because of the common error in estimated vehicle location [7]. The implication of this was profound: A consistent full solution to the combined localisation and mapping problem would require a joint state composed of the vehicle pose and every landmark position to be updated following each landmark observation. This would require the estimator to employ a huge state vector (on the order of the number of landmarks maintained in the map) with computation scaling as the square of the number of landmarks. Crucially, this work did not look at the convergence properties of the map or its steady-state behaviour. Indeed, it was widely assumed that the estimated map errors would not converge and would instead exhibit random-walk behaviour with unbounded error growth. Thus, given the computational complexity of the mapping problem and without knowledge of the convergence behaviour of the map, researchers focused instead on a series of approximations to the consistent mapping problem. They assumed or even forced the correlations between landmarks to be minimised or eliminated. Thus, reducing the full filter to a series of decoupled landmark to vehicle filters ([8-9] for example). This lead to the theoretical work on the combined localisation and mapping problem came to be temporarily halted, with work often focused on either mapping or localisation as separate problems.

The conceptual breakthrough came with the realisation that the combined mapping and localisation problem once formulated as a single estimation problem was actually converging.

Most importantly, it was recognised that the correlations between landmarks, which most researchers had tried to minimise, were actually the critical part of the problem and that, on the contrary, when the correlations grow, the solution be better. The structure of the SLAM problem, the convergence result and the coining of the acronym SLAM was first presented in a mobile robotics survey paper at the International Symposium on Robotics Research in 1995 [10]. The essential theory on convergence and many of the initial results were developed by Csorba [11-12]. Several groups already working on mapping and localisation, notably at the Massachusetts Institute of Technology [13], Zaragoza [14-15], the ACFR at Sydney [16-17] and others

[18-19], began working deeply on SLAM—also called *concurrent mapping and localisation* (CML) at that time—for indoor, outdoor, and subsea environments.

The International Symposium on Robotics Research in 1999 [20], was an important meeting for the field. In this symposium the first SLAM session was held and a degree of convergence between the Kalman-filter-based SLAM methods and the probabilistic localisation and mapping methods introduced by Thrun [21] was achieved. In 2000, at the IEEE International Conference on Robotics and Automation (ICRA), a Workshop on SLAM attracted 15 researchers and focused on issues such as algorithmic complexity, data association, and implementation challenges.

The following SLAM workshop at ICRA 2002 attracted 150 researchers with a broad range of interests and applications. In 2002, the SLAM summer school hosted by Henrik Christiansen at KTH in Stockholm attracted all the key researchers together with 50 PhD students from around the world and was a tremendous success in building the field. Interest in SLAM has grown exponentially in recent years, and workshops continue to be held at both ICRA and IROS. The SLAM summer school ran in 2004 in Toulouse then in Oxford in 2006, where many papers were published on the inertial SLAM [22-24].

Solving SLAM with monocular or stereo vision systems is a crucial open area of research which addresses many autonomous navigation problems in real life applications. Davison, in [71], proposed a vision-based SLAM approach, which used active stereo head and odometry sensing to estimate the location of a robot in planar regions. In [94], Davison has looked at the localisation and mapping problem using data from a single passive camera and combined with an EKF filter. Nir and Bruckstein [95], proposed a particle filter based SLAM approach to estimate camera (vehicle) poses.

Currently, the Visual SLAM becomes a very challenging axe of research and many papers were published in [25-28]. Moreover, with the development of Aerial Vehicles, the Airborne SLAM is seen as an essential area of research and many research works on Real time, Airborne and Visual SLAM are investigated in [22, 24, 28-31] .

This brief summary of the SLAM state of art shows that the Visual Airborne SLAM is still an open and challenging area of research to be yet fully explored.

1.3 Research Motivation

The main motivation behind this research, as the introduction has suggested, is the autonomous navigation of aerial vehicles. Today, the navigation of aerial vehicles becomes a challenging axe of research, firstly, because the airborne have 6 degrees of freedom, which make it more complicated than the mobile robot; secondly, the massive number of applications of aerial vehicle in military (surveillance, reconnaissance, battle damage assessment…) and civilian (farming, fire extinction…) applications.

Moreover, solving the airborne navigation problem using visual data is a new subject of research, which still under development. Thus, to investigate such a subject is a real challenging task because the camera is able to give the most complete information about the surroundings, which will be used not only to improve the airborne localisation but also to build a map of the environment. Furthermore, using invariant feature extractors to detect and match features is very important to construct a reliable map in natural environment.

One important aspect of the autonomous navigation which should be investigated as well is the fusion of data from different sensor. The data fusion algorithms are very important and their performances are closely dependant of both performances of the constructed map and the accuracy of the UAV position within this map. Optimal and robust filter is implemented and validated using real data.

The single airborne VSLAM using robust filter is a challenging task especially when it is validated using experimental data. To the best of my knowledge only few researchers are investigating this area. More than that, using multiple airborne VSLAM to explore and map a large area of navigation all we maintain an acceptable localisation accuracy still a real challenge to be attended particularly when it will be validated using experimental data.

1.4 Thesis Contributions and Organisation

This thesis focuses on developing a robust solution for the VSLAM problem to increase autonomy of Unmanned Aerial Vehicles. The investigated topics are related to data fusion, localisation, features extraction and matching, Map building, 3D pose

estimation and VSLAM. In addition, a cooperative VSLAM (C-VSLAM) solution is developed for a swarm of cooperating UAVs.

Most parts of this work have been either published or are under preparation for publication at highly ranked conferences and journals. The following summarises the contributions of the thesis highlighting the parts that have been written in a form of separate manuscript – all of which are listed below.

A brief summary of the contribution presented in this thesis are as follows:

• Robust INS/GPS data fusion (paper1)

In chapter (3), a robust approach for data fusion is proposed based on the nonlinear $H\infty$ ($NH\infty$) filter. The original contribution of this work is the proposition and the adaptation of the "Nonlinear $H\infty$ filter" to solve the INS/GPS UAV navigation problem. The proposed method solves issues related to linearisation, which is mostly linked to classical filtering techniques. Simulation results for 3D flight scenario are presented to demonstrate the advantages of the $NH\infty$ compared with results of classical EKF based technique. Good results were obtained with the $NH\infty$ filter without any assumption on the process and/or measurement noise characteristics, particularly in the case of strong nonlinearities or non centred noises.

• SDRE filter stability (paper3)

In chapter (3) a new INS/GPS sensor fusion scheme, based on State-Dependent Riccati Equation (SDRE) nonlinear filtering, for Unmanned Aerial Vehicle (UAV) localisation problem is presented. SDRE navigation filter is proposed as another alternative to Extended Kalman Filter (EKF), which has been extensively used in the literature. Based on optimal control theory, the SDRE filter solves issues linked with EKF filter such as linearisation errors that severely decrease UAV localisation performances. Proofs of stability of the SDRE nonlinear filter that are missing in the literature are also presented and validated on a 3D UAV flight scenario. Results obtained by SDRE navigation filter were compared to EKF navigation filter results. This comparison shows a better UAV localisation performance using SDRE filter.

• Feature extraction (ASIFT) **(paper4)**

In chapter (4) a novel technique to design a robust feature extractor and descriptor for visual map building is proposed. The extracted features are required to be computationally attractive and invariant to image rotation, scale change and illumination. We adapted the Scale Invariant Features Transform (SIFT) algorithm for map building applications. Our main contributions are: firstly, we introduce an adaptive version of the SIFT algorithm suitable for different visual perceptual environments. Secondly, we use the L-infinity norm as a criterion for feature matching, which ensures more robustness against noises and uncertainties. Finally, we propose a new criterion to select the most stable features in order to improve the visual map building performances. Results based on real images shows the good performance obtained with the proposed approach.

• Robust Map building and Super-resolution **(paper5)**

In the second part of chapter (4) an efficient, robust and automated frame registration and mosaicing algorithms which may be applied to real image sequences is developed. Image registration is the key step in combining multiple independent low-resolution images to give a one large mosaic image with high resolution. Fundamentally, this problem requires knowing which parts of the image to join up. In this work, we propose automatic techniques based on Adaptive Scale Invariant Features Transform (ASIFT detector/descriptor) for geometric and photometric image registration, mosaicing and super-resolution. Thus our approach is insensitive to the ordering, orientation, scale and illumination of the images.

• A novel Feature Extractor **(paper6)**

In the third part of chapter (4) a novel robust and fast feature detector and descriptor called Scale Wavelet Feature Invariant Transform (SWIF) is proposed. The SWIF algorithm is based on the Multi wavelet decomposition and look for features in scale and space following three directions. Scale Wavelet Invariant Feature (SWIF) detector/descriptor seems to be a promising solution for the VSLAM problem. Its

robustness using a short descriptor makes the difference in comparison to the proposed feature extractors available in the literature.

• Robust 3D VSLAM (paper2, paper7)

In chapter (5) a robust airborne 3D VSLAM solution based on a stereovision system is presented. Three innovative contributions to the Airborne VSLAM are proposed. The first one is the development of an alternative data fusion nonlinear $H\infty$ filtering scheme. This scheme is based on 3D vision observation model and avoids issues linked with the classical EKF techniques such as the linearisation errors, the initialisation problem and noise statistics assumptions. The second contribution consists of the consistency and the observability analysis for the Airborne VSLAM. The third contribution is a new approach to map management, based on the *k*-nearest landmark concept, and allowing efficient loop closure detection and map building. This approach reduces considerably the complexity of our Airborne VSLAM algorithm, which becomes independent of the map landmark number. Simulation results show the efficiency of the proposed Airborne VSLAM solution comparing with other techniques.

• Experimental 3D VSLAM in unknown environments (paper8)

The second part of chapter (5) presents a robust approach to solve the airborne VSLAM problem based on the development of a full stereo camera observation model. An adaptive SIFT feature extractor and an $L\infty$ norm distance criterion for robust feature detection and matching, respectively, were introduced. Robust 3D VSLAM is implemented based on the $NH\infty$ filter and compared with the EKF VSLAM while a new map management approach using the nearest *k*-landmarks criterion to reduce the algorithm complexity is validated. The proposed airborne VSLAM solution was validated using real data. Good and very promising results were obtained

• Robust Cooperative VSLAM (paper9)

Chapter (6) presents a framework for Cooperative Aerial VSLAM (C-VSLAM). The use of cooperative vehicles shows many advantages over single-vehicle architecture. We present the nonlinear H∞ CVSLAM based on extending our robust single vehicle VSLAM introduced in chapter (4). Loop closure concept, based on

revisited features is described with feature uncertainty analysis. Comparisons between single and multiple UAV VSLAM are made using realistic simulation scenarios.

• **Experimental Cooperative VSLAM** **(paper10)**

In the second part of Chapter (6), an experimental validation of C-VSLAM is presented. Full stereo camera observation model, Adaptive SIFT feature extractor and stereo vision constraints fast matching were introduced to construct a reliable large map. The Robust C-VSLAM with map management and loop closing is validated with simulation and using experimental data. Good and very promising results were obtained.

(1) **Abdelkrim Nemra** and Nabil Aouf "Robust nonlinear filtering for INS/GPS UAV localisation" IEEE proceeding, 16th, MED Conference on Control and Automation, 695-702, France, Ajaccio, June 2008.

(2) ML. Benmessaoud, A. Lamrani, **A. Nemra**, and AK. Souici "Single-Camera EKF-*v*SLAM" International Journal of Intelligent Systems and Technologies, vol 3, Num 4, WASET, 226-231, Paris, June 2008.

(3) **Abdelkrim Nemra** and Nabil Aouf "Robust INS/GPS Sensor Fusion for UAV Localisation using SDRE Nonlinear Filtering" IEEE Sensors Journal, Vol 10, N 4, January 2009.

(4) **Abdelkrim Nemra** and Nabil Aouf "Robust Feature Extraction and Correspondence for UAV Map Building" IEEE proceeding, 17th, MED Conference, Greece, June 2009.

(5) **Abdelkrim Nemra** and Nabil Aouf "Robust Invariant Automatic Image Registration, mosaicing and super resolution", IEEE proceeding, 6th International Symposium on Mechatronics and its Applications, Emirate, February 2009.

(6) **Abdelkrim Nemra** and Nabil Aouf "A novel fast and robust feature extractor and descriptor", to be Submitted.

(7) **Abdelkrim Nemra** and Nabil Aouf "Robust Airborne 3D Visual Simultaneous Localisation and Mapping with Observability and Consistency Analysis" Journal of Intelligent and Robotic Systems, Springer Science, DOI 10.1007/s10846-008-9306-6, January 2009.

(8) Abdelkrim Nemra and Nabil Aouf "Robust Cooperative UAVs Visual SLAM", IEEE Systems, Man and Cybernetics, United Kingdom and Republic of Ireland Chapter, UK, Birmingham, September 2009.

(9) Abdelkrim Nemra and Nabil Aouf "Experimental airborne NH∞ vision-based simultaneous localisation and mapping in unknown environments" JAERO805, Proc. IMechE Vol. 224 Part G: J. Aerospace Engineering April 2010.

(10) Abdelkrim Nemra and Nabil Aouf "Experimental Robust Cooperative Visual SLAM in Natural environment", under review in IEEE Intelligent Transportation Systems Transactions.

Chapter 2

UAV Sensors

CHAPTER 2

UAV sensors

2.1 Overview

In this chapter, we present the most popular sensors used by unmanned aerial vehicles. The chapter is divided on three sections. In the first section the Inertial Navigation System (INS) model and the errors that may affect it are presented. Then the Global Positioning System (GPS) is proposed as a complimentary sensor for the INS in long term navigation. Finally, visual cameras are presented as new alternative sensors for UAV navigation.

2.2 Introduction

In airborne applications, navigation systems can generally be divided into two categories: inertial (or dead-reckoning) navigation, and reference (or absolute) based navigation. An inertial navigation system (INS) makes use of an Inertial Measurement Unit (IMU) to sense the vehicle's rotation rates and accelerations. This data is then used to obtain vehicle states such as position, velocity and attitude at high sampling rates which are crucial for airborne guidance and control. However, INS diverging errors due to the integration process, requires absolute sensors in order to constrain these drifts.

Absolute sensors can be further categorised into two groups: beacon based or terrain based. The most common beacon based navigation system is the Global Navigation Satellite System (GNSS). There have been extensive research activities in the fusion of INS and GNSS systems [32-35]. The GNSS aided inertial navigation system provides long-term stability with high accuracy and has worldwide coverage for any weather condition. The main drawback is its dependency on external satellite signals, which can be blocked or jammed by intentional interferences.

Another sensor, which has been emerging very quickly in the last decade for autonomous vehicles and more specifically airborne vehicle, is the camera. This latter is available with different characteristics and low prices. Furthermore, cameras provide images, which are very useful to construct a map of the environment. These advantages

of cameras make computer vision as an essential research area for autonomous system applications.

2.3 Inertial Navigation Systems

INS consists of 3-axis gyroscopes, which provide to the system computer roll, pitch and yaw rates about the body. It also has 3-axis accelerometers, which provide accelerations along the three body axes. There are two basic inertial mechanisms, which are used to derive the Euler angles from the rate gyros: stable platform and strap-down INS. We would focus on strap-down INS, where gyros and accelerometers are `strapped-down' to the aircraft body frame. The acceleration values from the accelerometers are then corrected for rotation of the earth and gravity to give the velocity and position of the aircraft.

The localisation problem of an airborne system has been formulated based on the core-sensing device, an IMU, Figure (2.1). This unit measures the accelerations *(a_x, a_y, a_z)* and the rotation rates *(p, q, r)* of the airborne platform with high update rates, which can then be transformed and processed to provide its position (X, Y, Z), velocity (U, V, W) and attitude (ϕ, θ, ψ) resulting in an Inertial Navigation System (INS), as illustrated in Figure (2.1).

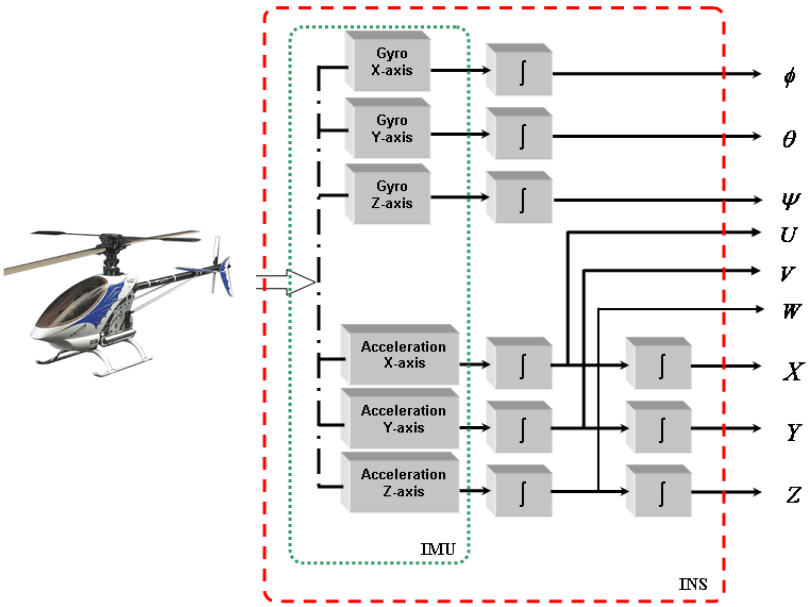


Figure.2. 1 INS architecture

2.3.1 INS Model

Let us present the INS with the following nonlinear model

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = h(x(t), u(t), t) \end{cases} \quad \textbf{(2.1)}$$

Where x is the state vector, which contains the position, velocity and Euler angles, and u represents the IMU outputs (angular rates and accelerations).

$$x = [\, X\,,\ Y\,,\ \ Z\,,\ U\,,\ \ V\,,\ W\,,\ \ \phi\,,\ \ \theta\,,\ \ \psi\,]^T \quad \textbf{(2.2)}$$

$$u = [\, p,\ \ q,\ \ r,\ \ \ ax,\ \ \ ay,\ \ \ az]^T \quad \textbf{(2.3)}$$

The navigation equations require defining at least two frames. One for the body/inertial representation (vehicle), and one for the navigation frame representation. Then, the equations of motion can be given by a simple integrations and frame transformations.

2.3.2 Equations of Motion

The Euler angle rates $\dot{\phi},\ \dot{\theta},\ \dot{\psi}$ can be calculated using:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \textbf{(2.4)}$$

Assuming that the IMU is at the vehicle centre of gravity, the true vehicle acceleration in the body frame is given by $\dot{U},\ \dot{V}, \dot{W}$:

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} ax + Vr - Wq + g\sin(\theta) \\ ay - Ur + Wp - g\cos(\theta)\sin(\phi) \\ az + Uq - Vp - g\cos(\theta)\cos(\phi) \end{bmatrix} \quad \textbf{(2.5)}$$

The resulting acceleration vector is integrated with respect to time to obtain the velocity of the vehicle in the body frame:

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \int \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} dt \tag{2.6}$$

The velocity vector is then integrated to read the position of the vehicle in the body frame. If the velocity is transformed down to navigation frame and integrated, we get the position $[X, Y, Z]^T$ vector in the navigation frame.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \int C^T{}_{bn}(\phi, \theta, \psi) \begin{bmatrix} U \\ V \\ W \end{bmatrix} dt \tag{2.7}$$

Where C_{bn} is the Direct Cosine Transform matrix that rotates a vector from body frame to the navigation frame.

$$C_{bn} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ \sin(\theta)\sin(\phi)\cos(\psi) - \sin(\psi)\cos(\phi) & \sin(\psi)\sin(\theta)\sin(\phi) + \cos(\psi)\cos(\phi) & \sin(\phi)\cos(\theta) \\ \sin(\theta)\cos(\phi)\cos(\psi) + \sin(\psi)\sin(\phi) & \sin(\phi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{2.8}$$

Then, the Nonlinear INS process state model is given by Equation (2.9).

$$f(x,u) = \begin{bmatrix} \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ \sin(\theta)\sin(\phi)\cos(\psi) - \sin(\psi)\cos(\phi) & \sin(\psi)\sin(\theta)\sin(\phi) + \cos(\psi)\cos(\phi) & \sin(\phi)\cos(\theta) \\ \sin(\theta)\cos(\phi)\cos(\psi) + \sin(\psi)\sin(\phi) & \sin(\phi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix}^T \begin{bmatrix} U \\ V \\ W \end{bmatrix} \\ \begin{bmatrix} ax + Vr - Wq + g\sin(\theta) \\ ay - Ur + Wp - g\cos(\theta)\sin(\phi) \\ az + Uq - Vp - g\cos(\theta)\cos(\phi) \end{bmatrix} \\ \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \end{bmatrix} \tag{2.9}$$

The observation model can be given by (2.10) when the aircraft position is measured by an absolute sensor (GNSS).

$$h(x,u)=\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\end{bmatrix} \quad \textbf{(2.10)}$$

The navigational solution provided by INS drifts with time, as in most other dead reckoning systems. The drift rate of the inertial position is typically a cubic function of time, which makes the development of any inertial based localisation a big challenge. Even small errors in gyros will be accumulated in angle estimates (roll and pitch), which in turn misrepresent gravitational acceleration as the vehicle acceleration, resulting in quadratic velocity (and cubic position) errors. The following section presents the most dominant errors affecting the INS sensor.

2.3.3 Errors in the INS

Most INS errors attributed to the inertial are the residual errors exhibited by the installed gyros and accelerometers following calibration of the INS. The most dominant error sources are shown in Table (2.1).

Alignment errors	Roll, pitch and heading errors
Accelerometer bias or offset	A constant offset in the accelerometer output that changes randomly after each turn-on
Nonorthogonality of gyros and accelerometers	The axes of accelerometer and gyro uncertainty and misalignment
Gyro drift or bias (due to temperature changes)	A constant gyro output without angular rate presence
Gyro scale factor error	Result in an angular rate error proportional to the sensed angular rate
Random noise	Random noise in measurement

Table.2.1 Sensor generated errors in the INS

Errors in the accelerations and angular rates lead to steadily growing errors in position and velocity components of the aircraft due to integration. These are called navigation errors and there are nine of them - three position errors, three velocity errors, two attitude errors and one heading error. If an unaided INS is used, these errors grow with

time. For this reason, the INS is usually aided with either Global Positioning System (GPS), Doppler heading sensor or air-data dead reckoning systems.

2.4 Global Positioning System

GPS uses a one-way ranging technique from the GPS satellites that are also broadcasting their estimated positions. Signals from (at least) four satellites are used with the user generated replica signal and the relative phase is measured. Using triangulation the location of the receiver is fixed as illustrated in Figure (2.2.a). The GPS receiver coupled with the receiver computer returns elevation angle and azimuth angle between the user and satellite, measured clockwise positive from the true north, geodetic latitude and longitude of the user. The GPS ranging signal is broadcast at two frequencies: a primary signal at 1575.42 MHz (L1) and a secondary broadcast at 1227.6 MHz (L2). Civilians use L1 frequency which has two modulations, viz. C/A or Clear Acquisition (or Coarse Acquisition) Code, and P or Precise or Protected Code. C/A is unencrypted signal broadcast at a higher bandwidth and is available only on L1. P code is more precise because it is broadcast at a higher bandwidth and is restricted for military use. The military operators can degrade the accuracy of the C/A code intentionally and this is known as Selective Availability.

There are six major causes of ranging errors: satellite ephemeris, satellite clock, ionospheric group delay, tropospheric group delay, multi-path and receiver measurement errors, including software. Ephemeris errors occur when the GPS message does not transmit the correct satellite location and this affects the ranging accuracy. These tend to grow with time from the last update of the control station. Satellite clock errors affect both C/A and P code users and leads to an error of 1-2m over 12hr updates [36]. Measurement noise affects the position accuracy of GPS pseudo range absolute positioning by few meters. The propagation of these errors into the position solution can be characterised by a quantity called Dilution of Precision (DOP), which expresses the geometry between the satellite and the receiver and is typically between 1 and 100. If the DOP is greater than 6, then the satellite geometry is not good. Ionospheric and tropospheric delays are introduced due to the atmosphere and this leads to a phase lag in calculation of the pseudo range. These can be corrected with dual-frequency P-code

receivers. Multi-path errors are caused by reflected signals entering the front end of the receiver and masking the correlation peak. These effects tend to be more prominent due to the presence of reflective surfaces, where 15m or more in ranging error can be found in some cases.

If we have a perfect clock in both the satellite and the receiver and if the signal is not affected by noise then it there would be easy to calculate the receivers position Figure (2.2.a). Otherwise, an additional satellite is needed for time offset of the GPS receiver clock Figure (2.2.b).

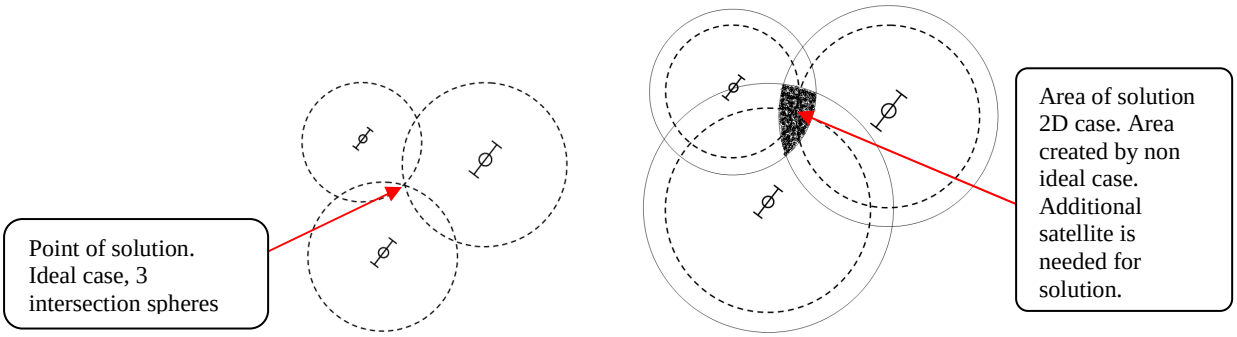


Figure.2. 2 GPS receiver localisation, a) Ideal case, 3 intersection spheres, b) Non ideal case, additional satellite is needed for time offset of GPS receiver clock

2.5 Visual system

Vision is the sensor, which is able to give the most complete information “what” and “where” of the objects an autonomous robot (aerial or ground) is likely to encounter. Humans are most certainly in possession of an active vision system. This means that we are able to concentrate on particular regions of interest in a scene by movements of the eyes and head or just by shifting attention to different parts of the images we see. With a particular goal in mind for an autonomous robot system, an active vision is able to select from the available information only what is directly relevant to seek the solution.

Figure.2. 3 left PTZ camera, right Bumblebee stereo-vision camera

Cheap cameras are available with different characteristics as illustrated in Figure (2.3). In our work we will use visible camera(s) to acquire visible images. Visible images are typically much sharper and clearer than infrared or thermal ones. Visible images present several advantages. First, visible images can be displayed in the same colors, shades, and intensities seen by the human eye, so their structure and character are more easily interpreted. Second, visible detector arrays have millions of elements, which is far more than those on an infrared detector. Finally, visible images are typically generated with reflected radiation. This can produce sharp contrasts and depict intensity differences: for example, a thin white line can be distinguished when it is next to a thin black line.

Although visible cameras have shown a powerful capacity of perception, which make them largely used in robotics navigation and mapping, they have a number of drawbacks. First, images brought by a camera need a large memory space that requires a significant computational time. Moreover, to extract regions of interest from an image powerful image processing algorithms should be proposed and implemented. Furthermore the problem of using a single camera is that it brings bearing information only, and not the depth (range) of the environment. As a solution the stereovision cameras are recently considered as the best way for 3D image reconstruction. Inspired from Human Visual System (HVS) stereovision cameras provide range and bearing of the environment which is very important for features localisation and map building. These advantages make stereo cameras a vital sensor to be used for airborne localisation and mapping.

2.6 Conclusion

In this chapter we presented the most important sensors used in aerospace navigation especially airborne localisation. As we explained each sensor has advantages and drawbacks, then a best result of airborne localisation can be obtained by fusion of data from two sensors or more. INS aided GPS is one common sensor fusion for long term airborne localisation. When the GPS signal is denied or when the map of the environment is as well required then the INS aided stereo cameras becomes the best solution for airborne navigation.

Chapter 3

UAV localisation

CHAPTER 3

UAV Localisation

3.1 Overview

In this chapter we investigate the UAV localisation problem by fusing INS and GPS data. First, both the popular Kalman and Extended Kalman filters are presented. Then, the optimal State Dependant Riccati Equation (SDRE) is proposed as an alternative, which doesn't require linearisation. After that the Nonlinear $H\infty$ filter ($NH\infty$) is implemented as a robust filter to face modeling error, process and observation noises. Finally, a comparison is made between the implemented filter using realistic UAV trajectories.

3.2 Introduction

Self-localisation is a crucial task for autonomous navigation, especially for airborne vehicles. The localisation of these latter becomes very difficult when the associated sensors are inaccurate. Although the localisation problem has been effectively solved in some standard settings, several issues are still open. In particular, localisation in 3D/dynamic environments, with incomplete or unknown maps.

INS errors presented in (Section 2.3.2 of chapter 2) make the development of any inertial only based localisation solution very unsuitable. Therefore, INS requires reliable and effective additional information to reduce these errors. The additional source, providing aerial vehicle position, adapted in this chapter is GPS. GPS signals are suitable to use in the sensor fusion process with the INS system by means of a navigation filtering scheme, as illustrated in Figure (3.1).

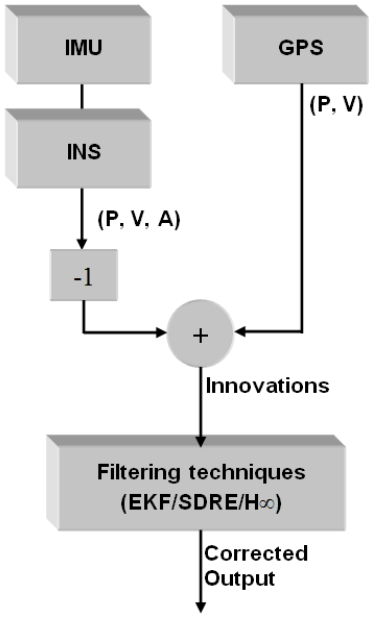


Figure.3. 1 INS aided GPS sensor fusion

In this chapter, few innovative filtering techniques are discussed; these techniques have been proposed to fuse INS/GPS data in order to solve the problem of UAV localisation.

3.3 Kalman Filter

The most popular filter in the literature is the famous Kalman filter proposed by Kalman in 1960 [37]. Kalman filter can be applied when the system is linear with Gaussian statistical distribution of the process and the observation noises. In this case, the system is completely described by Gaussian parameters as the mean and the covariance. Thus, the Bayesian prediction and update equation can be solved analytically [37]. As a discrete statistical recursive algorithm, Kalman filter provides an estimate of the state at time *k* given all observations up to time *k* and provides an optimal minimal mean squared error estimate of these states.

Kalman Filter (KF) is a very effective stochastic estimator for a large number of estimation, localisation and navigation problems. It is an optimal combination, in terms of variance minimisation, between the prediction of parameters from a previous time instant and external observations at a present time instant. In the following section, the Kalman filter formulation is presented details.

3.3.1 Process Model

A linear discrete time dynamic system can be described as:

$$x(k) = F_k x(k-1) + B_k u(k) + Gw(k) \quad \textbf{(3. 1)}$$

where $x(k)$ is the state vector of interest at time k. F_k is a linear state transition matrix which relates the state vector from step $k-1$ to k. $u(k)$ is the input control vector while B_k relates the control vector to the states. $w(k)$ is the process noise injected into the system due to uncertainties in the transition matrix and the control input while G relates the noise to the states.

The process noise w is assumed to be a zero mean, uncorrelated random sequence with covariance:

$$\begin{aligned} E[w(k)] &= 0 \qquad \forall k \\ E[w(k)w^T(j)] &= \begin{cases} Q(k) & k = j \\ 0 & k \neq j \end{cases} \end{aligned} \quad \textbf{(3. 2)}$$

Q is the covariance of the process noise.

3.3.2 Observation Model

When observations of the states are taken, the observation vector $z(k)$ at time k is given by:

$$z(k) = H_k x(k) + v(k) \quad \textbf{(3. 3)}$$

where H_k is the linear observation model relating the state vector at time k to the observation vector, and $v(k)$ is the observation noise vector which accounts for the uncertainty in the observation. The observation noise is also assumed to be a zero mean, uncorrelated random sequence with covariance:

$$E[v(k)] = 0 \quad \forall k$$
$$E[v(k)v^T(j)] = \begin{cases} R(k) & k = j \\ 0 & k \neq j \end{cases} \quad \textbf{(3. 4)}$$

R is the covariance of the process noise.

It is assumed that the process and observation noise are uncorrelated,

$$E[w(k)v(j)^T] = 0 \quad \forall k, j \quad \textbf{(3. 5)}$$

Given the process and observation models, Kalman filter provides a recursive estimate of the states at time k, $x_{(k \mid k)}$, given all observations up to time k.

3.3.3 Prediction

The predicted state is evaluated by taking expectations of Equation (3.1) upon the previous $k-1$ observations, without noise:

$$\hat{x}_{k/k-1} = F_k x_{k-1/k-1} + B_k u(k-1) \quad \textbf{(3. 6)}$$

The uncertainty in the predicted states at time k, $P_{k/k-1}$, is described as the expected value of the variance of the error in the states at time k given all information up to time $k-1$,

$$P_{k/k-1} = F_k P_{k-1/k-1} F_k^T + G.Q_k.G^T \quad \textbf{(3. 7)}$$

3.3.4 Estimation

When an observation from an external aiding sensor is obtained as in Equation (3.3), an estimate of the state is obtained by,

$$x_{k/k} = x_{k/k-1} + K(z_k - H_k x_{k/k-1}) \quad \textbf{(3. 8)}$$

where K, a gain matrix (Kalman gain) is produced by the Kalman filter and $(z_k - H_k x_{k/k-1})$ is the innovation vector. The innovation vector is the difference between the actual observation and the predicted observation. The predicted observation is determined by taking the expected value of Equation (3.3) conditioned on previous observations. Equation (3.8) defines the update as simply as the latest prediction plus a weighting on the innovation. The Kalman gain *K* or weighting is chosen so as to minimise the mean squared error of the estimate,

$$K = P_{k/k-1} H_k^T S_k^{-1} \quad \textbf{(3. 9)}$$

where S_k is known as the innovation covariance and is obtained by,

$$S_k = H_k P_{k/k-1} H^T + R_k \quad \textbf{(3. 10)}$$

The covariance matrix, or the uncertainty in the updated states, is obtained by taking the expectation of the variance of the error at time *k* given all observations up to time *k*,

$$P_{k/k} = [I - KH_k] P_{k/k-1} \quad \textbf{(3. 11)}$$

We can also write

$$P_{k/k} = [I - KH_k] P_{k/k-1} [I - KH_k]^T + KR_k K^T \quad \textbf{(3. 12)}$$

Equation (3.12) is called the Joseph form of the covariance update that assures the symmetry and positive definiteness of $P_{k/k}$ [37].

3.4 Extended Kalman Filter

In most real applications the process and/or observation models are nonlinear and hence the linear Kalman filter algorithm described above cannot be directly applied. To overcome this limitation, a linearized Kalman filter or Extended Kalman Filter (EKF) can be applied, where system models are continuously linearized around the estimation states before applying the estimation techniques [37, 38]. In some applications, a predetermined nominal trajectory to navigate on exists. In this case, the nonlinear state model can be linearized around the nominal trajectory and linear Kalman filter theory can be used. The filter gain, which is computationally expensive can also be computed off-line and can be used as a look-up table in real-time operation.

However, in most practical navigation applications, a nominal trajectory does not exist beforehand. The solution is then to use the current estimated state from the filter at

each time step *k* as the linearisation reference from which the estimation procedure can proceed. Such an algorithm is known as the Extended Kalman Filter. If the filter operates properly, the linearisation error around the estimated solution can be maintained at a reasonably small value. However, if the filter is ill-conditioned due to modelling errors, incorrect tuning of the covariance matrices, or initialisation error, then the estimation error will affect the linearisation error. This latter, in turn, will affect the estimation process and is known as filter divergence. For this reason the EKF requires greater care in modelling and tuning than the linear Kalman filter.

The system as defined in Equation (2.1), is written as a non-linear discrete time state transition equation:

$$\begin{aligned} x_k &= f(x_{k-1}, u_k, w_k) \\ y_k &= h(x_k, v_k) \end{aligned} \quad \textbf{(3. 13)}$$

x_k is the state at time step k, u_k control vector at time step k, w_k is some additive noises, y_k is the observation made at time k, v_k is some additive observation noises. We assume that w_k and v_k are uncorrelated zero mean Gaussian with known covariance Q_k and R_k. The objective of the filtering technique is, then, to estimate x_k using available observation y_k.

The non-linear vehicle model and observation model maybe expanded about the filtered and predicted estimates of $\hat{x}_k$ and $\hat{x}_{k-1}$.

$$x_k = f(\hat{x}_{k-1/k-1}, u_k) + \nabla f_k(x)[x_k - \hat{x}_{k/k}] + \Delta_1(x_k - \hat{x}_{k/k}) + [\nabla f_w(x) + \Delta_2(x_k - \hat{x}_{k/k})]w_k \quad \textbf{(3. 14)}$$

$$y_k = h(\hat{x}_{k/k-1}, u_k) + \nabla h_k(x)[x_k - \hat{x}_{k/k-1}] + \Delta_3(x_k - \hat{x}_{k/k-1}) + v_k \quad \textbf{(3. 15)}$$

Where $\nabla f_k(x)$ is the Jacobian of f evaluated at x_{k-1}, $\nabla f_w(x)$ the Jacobian of f / w_k evaluated at x_{k-1} and $\nabla h_k(x)$ is the Jacobian of h evaluated at x_{k-1} and Δ_i represent higher order of the Taylor series expansions. These higher order terms are norm bounded as $\|\Delta_i\| \le \delta_i$.

The filter state error is defined

$$\tilde{x}_{k/k} = x_k - \hat{x}_{k/k} \quad \textbf{(3. 16)}$$

The prediction error can be determined from subtracting the true state x_k from the prediction estimate.

$$\tilde{x}_{k/k-1} = x_k - \hat{x}_{k/k-1} \quad \textbf{(3. 17)}$$

The state and observation model may be rewritten as:

$$\begin{aligned} x_{k+1} &= F_k x_k + \Gamma_k w_k + \Omega_k + \Delta_1(\tilde{x}_{k/k}) + \Delta_2(\tilde{x}_{k/k}) w_k \\ y_k &= H_k x_k + v_k + \Psi_k + \Delta_3(\tilde{x}_{k/k-1}) \end{aligned} \quad \textbf{(3. 18)}$$

with $F_k = \nabla f_k(\hat{x}_{k/k}), \Gamma_k = \nabla f_w(\hat{x}_{k/k}),\ H_k = \nabla h_k(\hat{x}_{k/k-1}),\ \Omega_k = f(\hat{x}_{k/k}) - F_k \hat{x}_{k/k}$ and $\Psi_k = h(\hat{x}_{k/k-1}) - H_k \hat{x}_{k/k-1}$

Then, the final EKF formulation is written in Predictor-Corrector scheme with the higher order terms of the Taylor series equal to zero.

Predictor:

$$\hat{x}_{k+1/k} = f(\hat{x}_{k/k}, u_k, 0) \quad \textbf{(3. 19)}$$

$$P_{k+1/k} = F_k P_{k/k} F_k^{\ T} + \Gamma_k Q_k \Gamma^T{}_k \quad \textbf{(3. 20)}$$

Corrector:

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + K_k(y_k - H_k x_{k/k-1}) \quad \textbf{(3. 21)}$$

$$K_k = P_{k/k-1} H_k (H_k P_{k/k-1} H^T{}_k + R_k)^{-1} \quad \textbf{(3. 22)}$$

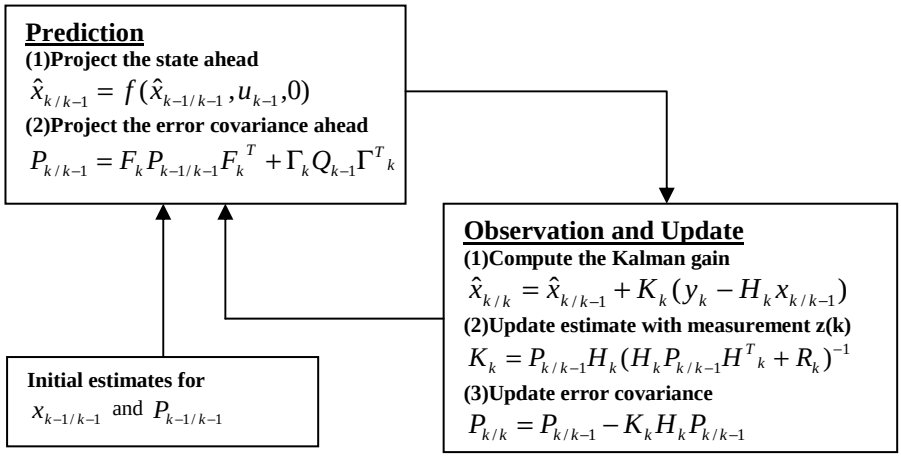
$$P_{k/k} = P_{k/k-1} - K_k H_k P_{k/k-1} \quad \textbf{(3. 23)}$$

Or using the Joseph form

$$P_{k/k} = (I - K_k H_k) P_{k/k-1} (I - K_k H_k)^T + K_k \nabla h_v(k) R_k \nabla h_v^{\ T}(k) \quad \textbf{(3.23')}$$

$\nabla h_v(k)$ is the Jacobian with respect to the observation noise **v**(k).

Below a flowchart showing the main steps of the EKF algorithm:



Extended Kalman Filter algorithm

When there are large deviations of the estimated state trajectory from the nominal trajectory, the nonlinear model is weakly approximated by the Taylor series expansion about the conditional mean. In this case, the higher order terms of the Taylor expansion become necessary. In EKF, the higher order terms are neglected. In the following section, we propose other alternatives to avoid the linearisation problem.

3.5 Optimal Filters

To avoid the linearisation problem, many data fusion techniques based on probabilistic approaches were presented and used in the literature. One of these techniques is Particle Filter (PF) [134-135]. The main drawback of this filter is its computational requirement, which makes it not very suitable for real time applications such as aerial navigation problem. Approaches based on Unscented Transform (UT) resulted in a technique called Unscented Kalman Filter (UKF) [136]. This technique preserves the linear update structure of Kalman filter. It uses only second order system moments, which may not be sufficient for some nonlinear systems. In addition, the number of sigma points, used in UKF, is small and may not represent adequately complicated distributions. Moreover, unscented transformation of the sigma-points is computationally heavy, which is not suitable and practical for real time aerial navigation applications.

In the following section we investigate an alternative to EKF based data fusion technique for UAV localisation problem. This alternative is based on INS and GPS data and uses a State-Dependent Riccati Equations (SDRE) non-linear filtering formulation.

3.5.1 SDRE Nonlinear filter

State-dependent Riccati equation (SDRE) techniques are rapidly emerging as general design methods, which provide a systematic and effective means of designing nonlinear controllers, observers, and filters. Numerous design methodologies exist for the control design of highly nonlinear systems. These include any of the huge number of linear design techniques [39-42] used in conjunction with gain scheduling [43-45], nonlinear design methodologies such as dynamic inversion [46], sliding mode control [46], recursive back-stepping [47]; and adaptive techniques which encompass both linear adaptive [48] and nonlinear adaptive [47] control. Less known nonlinear design procedures are those that involve state-dependent Riccati equations (SDRE) [49, 50]. Over the past several years various SDRE design methodologies have been successfully applied to aerospace problems. SDRE-based design procedures have been used in advanced guidance law development [51, 52], in an output feedback (nonlinear *H2)* autopilot design [53], and in a full information nonlinear $H\infty$ autopilot design [54]. In an earlier related work, a parameter-dependent Riccati equation was used in a pitch-yaw autopilot design where the parameter, roll rate, was exogenously supplied by the roll autopilot [55]. Additionally, SDRE design methods have been used in nonlinear filter development [50] and in control designs for other nonlinear benchmark problems [56, 57]. In [58], Ehrler and Vadali investigated the nonlinear regulator problem and showed that solving an algebraic Riccati equation, as it evolved over time, provided one means of obtaining a suboptimal solution of the infinite horizon problem. In essence, the state-dependent Riccati equation was treated as being time-dependent and its state dependency was not explicitly acknowledged, addressed, or analysed. In [49], SDRE nonlinear regulation, SDRE nonlinear $H\infty$, and SDRE nonlinear *H2* design methodologies were defined and the optimality, sub-optimality, and stability properties of SDRE nonlinear regulation were investigated. In [50], the SDRE nonlinear filter was defined and used to track a pendulum. In the following, this filtering design methodology will be presented and the supporting theory developed to date will be reviewed.

The SDRE nonlinear filter is based on the State Dependent Coefficient (SDC) factorisation [59]. From [60] and [61], it can be seen that the autonomous infinite-horizon nonlinear regulator problem, being considered, is a generalisation of the time-invariant infinite-horizon linear-quadratic regulator problem. In this problem, all the coefficient matrices are state-dependent. When these coefficient matrices are constant, the nonlinear regulator problem collapses to the linear regulator problem and the SDRE control method collapses to the steady-state linear regulator. We can obtain the filtering counterpart of the SDRE control algorithm by taking the dual of the steady-state linear regulator and then allowing the coefficient matrices of the dual to be state-dependent [61]. The dual of the steady-state linear regulator is the steady-state continuous Kalman observer. This latter reduces to the steady-state continuous Kalman filter [61], when the control is absent. This leads to the following:

3.5.2 State Dependent Coefficient (SDC) Form

Consider the continuous stochastic nonlinear system:

$$\begin{aligned} \dot{x} &= f(x) + \Gamma w \\ y &= h(x) + v \end{aligned} \qquad \textbf{(3. 24)}$$

w and v are Gaussian zero mean white noises with covariance matrices Q and R, respectively. Γ is a weighting matrix of the process noise. There are an infinite number of ways to transform this system into an SDC form:

$$\begin{aligned} \dot{x} &= F(x)x + \Gamma w \\ y &= H(x)x + v \end{aligned} \qquad \textbf{(3. 25)}$$

where $f(x) = F(x)x \qquad h(x) = H(x)x$

Note that the INS estimation model presented in Equation (2.9, 2.10) falls naturally, with equation manipulations, into an SDC form which make the SDRE filtering very appealing to use for this localisation based fusion problem.

3.5.3. SDRE Non-Linear Filter

The SDRE filter uses this SDC form and is given in [49] by Cloutier and al:

$$\dot{\hat{x}} = F(\hat{x})\hat{x} + K_f(\hat{x})[y(x) - H(\hat{x})\hat{x}] \qquad \textbf{(3. 26)}$$

where

$$K_f(\hat{x}) = PH^T(\hat{x})R^{-1} \quad \textbf{(3. 27)}$$

and where P is the positive definite solution to the algebraic Riccati equation:

$$F(\hat{x})P + PF^T(\hat{x}) - PH^T(\hat{x})R^{-1}H(\hat{x})P + \Gamma Q\Gamma^T = 0 \quad \textbf{(3. 28)}$$

Some of the properties of the SDRE methods are stated in [60] and the proofs of these properties are given in [59]. In addition, $f(x)$ and $h(x)$ are k-continually derivable (i.e. belonging to $C^k, k \geq 1)$. It is also assumed that the SDC parameterisation $F(x)$ and $H(x)$ are smooth $(i.e \in C^k, k \geq 1)$.

The SDRE nonlinear filter makes it possible to avoid the linearisation problem of the EKF. However, it is still based, not only on the restricted assumption of Gaussian noise statistics, but also on certain accuracy of the process and the observation model. Without these assumptions, the performance of this filter may be poor and lead to inaccuracy position. A solution to this problem is proposed in the next section (3.6).

3.6 The H∞ filter

Kalman and SDRE filter assume that the process has a known dynamics and that the exogenous inputs have known statistical properties. Unfortunately, these assumptions limit the utility of minimum variance estimators in situations where models and/or noises descriptions are unknown [62].

As we have seen, Kalman filter is an effective tool for estimating the states of nonlinear systems. The early success in the 1960s of the Kalman filter in aerospace applications led to attempt to apply it to more common other industrial applications in the 1970s. However, these attempts quickly showed that a serious mismatch existed between the underlying assumptions of Kalman filters and some industrial state estimation problems. In fact engineers rarely understand the statistical nature of the noise process that impinges on industrial processes. After long years of reappraising the nature and role of Kalman filters, engineers realised they needed new filters that could handle modeling errors and noise uncertainties. State estimators that can tolerate such uncertainty are called robust. Although robust estimators based on Kalman filter theory can be designed, these approaches are somewhat ad-hoc since they attempt to modify an already existing approach. $H\infty$ filters were specifically designed for robustness.

We will first derive an alternative form for the Kalman filter. This is done to facilitate comparisons that will made later in this paragraph between the Kalman and the H∞ filters. Then, the limitation of the Kalman filter will briefly be discussed.

3.6.1 An Alternate Form for the Kalman Filter

Recall that Kalman filter estimates the states of a linear dynamic system defined by the equations:

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x + v_k \end{aligned} \quad \textbf{(3. 29)}$$

Where F_k the transition matrix, H_k the observation matrix, $\{ w_k \}$ and $\{ v_k \}$ are stochastic processes with covariance Q_k and R_k, respectively. As shown before, the Kalman filter equations are given as follows:

$$\begin{aligned} \hat{x}_{k+1/k} &= F_k \hat{x}_{k/k-1} + F_k K_k (y_k - H_k \hat{x}_{k/k-1}) \\ K_k &= P_{k/k-1} H_k{}^T (H_k P_{k/k-1} H_k{}^T + R_k)^{-1} \\ P_{k/k-1} &= F_{k-1} P_{k-1/k-1} F^T{}_{k-1} + Q_{k-1} \\ P_{k/k} &= (I - K_k H_k) P_{k/k-1} \end{aligned} \quad \textbf{(3. 30)}$$

After developing and substituting, the Kalman filter can be rewritten in the following form [82]:

$$\begin{aligned} \hat{x}_{k+1/k} &= F_k x_{k/k-1} + F_k K_k (y_k - H_k \hat{x}_{k/k-1}) \\ K_k &= P_{k/k-1} (I + H^T{}_k R^{-1}{}_k H_k P_{k/k-1})^{-1} H^T{}_k R_k{}^{-1} \\ P_{k+1/k} &= F_k P_{k/k-1} (I + H_k{}^T R_k{}^{-1} H_k P_{k/k})^{-1} F_k{}^T + Q_k \end{aligned} \quad \textbf{(3. 31)}$$

3.6.2 Kalman Filter Limitations

Kalman filter works well but only under certain conditions:

- First, the mean and the correlation of the noise w_k and v_k should be known at each time instant.
- Second, the covariance Q_k and R_k of the noise processes should be known. Kalman filter uses Q_k and R_k as design parameters. Thus if Q_k and R_k are not known, then it may be difficult to successfully use Kalman filter.

- Third, the attractiveness of the Kalman filter lays in the fact that it gives: the best minimum variance estimator if the noise is white and Gaussian and it gives the best linear minimum variance estimator if the noise is white and not Gaussian. However if one of these assumption does not hold then Kalman filter can be applied but no optimal performance is guaranteed.
- Finally, the system model matrices F_k and H_k should be known with high accuracy.

What does happen if one of these Kalman filter assumptions is not satisfied? What should we do if we do not have any information about noise statistics? What should we do if we want to minimize the worst-case estimation error rather than the covariance of the estimation error?

May be the Kalman filter can be just used, even-though its assumptions are not satisfied, and hope for best. That is a common solution to the Kalman filter quandary and it works reasonably well in many cases. However, there is yet another alternative option that will be explored: the $H\infty$ filter**,** also called the **minmax** filter [82]. The $H\infty$ filter does not make any assumption about the noise, and it minimises the worst-case estimation error (hence the term minmax).

3.6.3 Discrete-time H^∞ Filter

1. Assume that system of equations is given by:

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x_k + v_k \end{aligned} \qquad \textbf{(3. 32)}$$

where w_k and v_k are noise terms. These noises may be random with possibly unknown statistics, or they may be deterministic. They may have a nonzero mean as well. Our goal is to estimate a linear combination of the states. That is, we want to estimate z_k which is given by:

$$z_k = L_k x_k$$

where L_k is a used defined matrix (assumed to be full rank), If we want to directly estimate x_k (as in the Kalman filter) then we set $L_k = I$.

2. The cost function is given as [82]:

$$J_1 = \frac{\sum_{k=0}^{N-1} \left\| z_k - \hat{z}_k \right\|^2_{S_k}}{\left\| x_0 - \hat{x}_0 \right\|^2_{P_0^{-1}} + \sum_{k=0}^{N-1} \left(\left\| w_k \right\|^2_{Q_k^{-1}} + \left\| v_k \right\|^2_{R_k^{-1}} \right)} \quad \textbf{(3. 33)}$$

where P_0, Q_k, R_k and S_k are symmetric, positive definite matrices chosen by the engineer based on the specific problem. Our goal is to find an estimate $\hat{z}_k$ that minimises J_1. However the direct minimisation of J_1 is not attractable, so instead we choose a performance bound and seek an estimation strategy that satisfies the threshold. That is, we will try to find an estimate $\hat{z}_k$ that result in:

$$J_1 < \frac{1}{\theta}$$

Where, θ is the user-specified performance bound [82]. Rearranging this equation results in:

$$J = -\frac{1}{\theta} \left\| x_0 - \hat{x}_0 \right\|^2_{P_0^{-1}} + \sum_{k=0}^{N-1} \left[\left\| z_k - \hat{z}_k \right\|^2_{S_k} - \frac{1}{\theta} \left(\left\| w_k \right\|^2_{Q_k^{-1}} + \left\| v_k \right\|^2_{R_k^{-1}} \right) \right] < 0 \quad \textbf{(3. 34)}$$

The minmax problem is then defined as:

$$J^* = \min_{\hat{z}_k} \max_{w_k, v_k, x_0} J$$

The worst-case is obtained when w_k, v_k and x_0 are chosen to maximise J. The solution then is to find an estimate $\hat{z}_k$ which minimises this maximum.

Since $y_k = H_k x_k + v_k$ then $v_k = y_k - H_k x_k$ and $\left\| v_k \right\|^2_{R_k^{-1}} = \left\| y_k - H_k x_k \right\|^2_{R_k^{-1}}$

Since $z_k = L_k x_k$ and $\hat{z}_k = L_k \hat{x}_k$ it is clear that:

$$\begin{aligned} \left\| z_k - \hat{z}_k \right\|^2_{S_k} &= (z_k - \hat{z}_k)^T S_k (z_k - \hat{z}_k) \\ &= (x_k - \hat{x}_k)^T L_k{}^T S_k L_k (x_k - \hat{x}_k) \\ &= \left\| x_k - \hat{x}_k \right\|^2_{\bar{S}_k} \end{aligned} \quad \textbf{(3. 35)}$$

where

$$\bar{S}_k = L_k{}^T S_k L_k \quad \textbf{(3. 36)}$$

Then we substitute these variables in Equation (3.34) to obtain:

$$J = -\frac{1}{\theta}\left\|x_0 - \hat{x}_0\right\|^2_{P_0^{-1}} + \sum_{k=0}^{N-1}\left[\left\|x_k - \hat{x}_k\right\|^2_{\bar{S}_k} - \frac{1}{\theta}\left(\left\|w_k\right\|^2_{Q_k^{-1}} + \left\|y_k - H_k x_k\right\|^2_{R_k^{-1}}\right)\right] < 0 \quad \textbf{(3. 37)}$$

3. The cost function J_1 can be made less than $1/\theta$ (a user-specified bound), based on the solution of the optimisation problem:

$$J^* = \min_{\hat{z}_k} \max_{w_k, v_k, x_0} J \quad \textbf{(3. 38)}$$

and leading to:

$$\begin{aligned} \bar{S}_k &= L_k{}^T S_k L_k \\ K_k &= P_k[I - \theta S_k P_k + H_k{}^T R_k^{-1} H_k P_k]^{-1} H_k{}^T R_k^{-1} \\ \hat{x}_{k+1} &= F_k \hat{x}_k + F_k K_k (y_k - H_k \hat{x}_k) \\ P_{k+1} &= F_k P_k [I - \theta S_k P_k + H_k{}^T R_k^{-1} H_k P_k]^{-1} F_k{}^T + Q_k \end{aligned} \quad \textbf{(3. 39)}$$

4. The following condition must hold at each time step k in order for the above estimator to be a solution to the optimisation problem :

$$P_k^{-1} - \theta S_k + H_k{}^T R_k^{-1} H_k > 0 \quad \textbf{(3. 40)}$$

3.6.4 $H\infty$ Filter Analysis

Before examining the full nonlinear *H*∞ Filter (*NH*∞) with respect to the minimum entropy *H*∞, we start by analysing the linear $H\infty$ filter. There exist two fundamental differences between the Kalman filter and H∞ filter. First, *H*∞ filter is optimal in terms of minimising the ∞-norm of the gain between a set of disturbance inputs, and the estimation error. Thus, the filter may be characterised by the fact that the worst-case gain is minimised. In contrast, Kalman filter minimises the mean square gain between the disturbances and the estimation error.

The second difference is that the minimum mean square estimate of Kalman filter commutes with linear operations. However, the minimal ∞-norm estimate does not possess this property and the *H*∞ optimal estimator depends on the plant output being estimated.

Using the ideas in Green and Limebeer [63] and Limebeer and al [64], we seek to find a linear filter that estimates the quantity:

$$z_k = C_k x_k \quad \textbf{(3. 41)}$$

From the observed data y_k such that the error

$$\tilde{z}_{k/k} = H_k x_k - \hat{y}_{k/k} \quad \textbf{(3. 42)}$$

Satisfies the worst-case performance criteria. The linear filtering problem is written

$$x_k = F_k x_{k-1} + \Gamma_k u_k + G w_k \quad \textbf{(3. 43)}$$

$$y_k = H_k x_k + v_k \quad \textbf{(3. 44)}$$

Where F_k is the transition matrix, Γ_k is a weighting matrix for the input u_k and G is a weighting matrix for the process noise.

The linear $H\infty$ filter Equation (3.39) may be placed in a predictor-corrector formulation as done in Kalman filter:

Predictor

$$P_{k+1/k} = F_k P_{k/k} F_k^{\ T} + \Gamma_k Q_k \Gamma_k^{\ T} \quad \textbf{(3. 45)}$$

$$\hat{x}_{k+1/k} = F_k \hat{x}_{k/k} + \Gamma_k u_k \quad \textbf{(3. 46)}$$

Corrector

$$\hat{x}_{k/k} = F_k \hat{x}_{k/k-1} + F_k L_k \tilde{z}_{k/k-1} \quad \textbf{(3. 47)}$$

$$L_k = P_{k/k-1} H_k^{\ T} (H_k P_{k/k-1} H_k^{\ T} + R_k)^{-1} \quad \textbf{(3. 48)}$$

$$P_{k/k} = P_{k/k-1} - P_{k/k-1}[-C_k^{\ T} \quad H_k^{\ T}] \Lambda^{-1}{}_{k/k} \begin{bmatrix} -C_k \\ H_k \end{bmatrix} P_{k/k-1} \quad \textbf{(3. 49)}$$

$$\Lambda_{k/k} = \begin{bmatrix} C_k P_{k/k-1} C_k^{\ T} - \gamma^2 I & -C_k P_{k/k-1} H_k^{\ T} \\ -H_k P_{k/k-1} C_k^{\ T} & H_k P_{k/k-1} H_k^{\ T} + R_k \end{bmatrix} \quad \textbf{(3. 50)}$$

With $\gamma^2 = \dfrac{1}{\theta}$

A minimum value for γ^2 must be found through searching over $\gamma \neq 0$ such that $P_{k/k} > 0$. When $\gamma = \infty$, $H\infty$ filter reverts to Kalman Filter. The γ can be used for tuning the trade off between $H\infty$ performance and the minimum variance performance. The Extended $H\infty$ filter ($EH\infty$) for the nonlinear system follows the same principle as the EKF based on the Taylor expansion of the nonlinear model of the filtered and predicted estimates.

3.6.5 Nonlinear H∞ Filter

Many researchers have studied nonlinear $H\infty$ optimal estimator such as, Shaked and Berman [65], Petersen and Savkin [66], Basar and Bernhard [67] and Einicke and White [68]. Our $H\infty$ filtering procedure uses a similar procedure as in Einicke and White [68]. The non-linear $H\infty$ attempts to estimate the states given in Equation (3.13) while satisfying the $H\infty$ filter performance criteria with respect to Δ_i and their norm bounds. The system given in Equation (3.18) can be rewritten into the following form:

$$\hat{x}_{k+1} = F_k \hat{x}_k + B_k w_k + \Omega_k + \Pi_k \quad \textbf{(3. 51)}$$

$$y_k = H_k x_k + v_k + \Psi_k + \sum_k \quad \textbf{(3. 52)}$$

where

$$\Pi_k = \Delta_1(\tilde{x}_{k/k}) + \Delta_2(\tilde{x}_{k/k})v_k \quad \textbf{(3. 53)}$$

and

$$\sum_k = \Delta_3(\tilde{x}_{k/k-1}) \quad \textbf{(3. 54)}$$

These inputs must satisfy the following norm bounds:

$$\|\Pi_k\|_2^2 \le \delta_1^2 \|\tilde{x}_{k/k}\|_2^2 + \delta_2^2 \|w_k\|_2^2 \quad \textbf{(3. 55)}$$

and

$$\|\Sigma_k\|_2^2 \le \delta_3^2 \|\tilde{x}_{k/k}\|_2^2 \quad \textbf{(3. 56)}$$

Instead of solving for the non-linear vehicle model and observation model containing the extra terms Π_k and Σ_k which are not used in the EKF, the following scaled $H\infty$ problem is considered [68]:

$$\hat{x}_{k+1} = F_k \hat{x}_k + B_k c_w w_k + \Omega_k \quad \textbf{(3. 57)}$$

$$y_k = H_k x_k + c_v v_k + \Psi_k \quad \textbf{(3. 58)}$$

Where $c_v^2 = 1 - \gamma^2\delta_1^2 - \gamma^2\delta_3^2$ and $c_w^2 = c_v^2(1+\delta_2^2)^{-1}$. This final form results in the same structure as the Extended $H\infty$ filter, except that the error covariance correction of the linear $H\infty$ filter Equation (3.48) is used with $C_k = 1$, and the noise process w_k and v_k scaled by c_w and c_v.

Notes:

- When $\delta_1 = \delta_2 = \delta_3 = 0$ the Nonlinear $H\infty$ reverts back to the Extended $H\infty$.
- As γ goes to ∞, the Extended $H\infty$ reverts back to the EKF.

Filters	**Advantages**	**Disadvantages**
KF	- Simple implementation - Optimal filter - Minimum variance estimator	- Linear filter - Noises should be centred, white and (Gaussian).
EKF	- Nonlinear filter - Simple to implement - Minimum variance estimator	- Linearisation errors - Noises should be centred, white and Gaussian.
SDRE	- Nonlinear filter - Do not require linearization. - Quasi optimal filter	- Noises should be centred, white and Gaussian. - Not suitable for complex system
NH∞	- Nonlinear filter - Robust filter - Does not make any assumption on noise characteristics.	- Model linearization is required - Minimising the ∞ norm (worst case)

Table 3. 1 Comparison between techniques

3.7 Results and Discussion

UAV localisation results are divided on two sections: First, we present and discuss a proof of stability of the SDRE nonlinear filter for localisation. In the second section, we present the simulation results of the implemented filters for the localisation problem. We begin by linear Kalman Filter (KF) and Extended Kalman Filter (EKF). Then we present the simulation work validating our proposed nonlinear SDRE filter for the autonomous airborne navigation problem. Results of the SDRE filter will be compared with other classical navigation filtering techniques. Finally, we present simulation results obtained by nonlinear H∞ filter.

3.7.1. SDRE Stability

Global stability of the SDRE nonlinear filter, as opposed to local stability of linear systems, is more difficult to prove since having stable eigenvalues for the discrete SDRE at sampling time does not guarantee global asymptotic stability. This makes stability analysis quite difficult and to the best of our knowledge no global stability formal proofs of the SDRE nonlinear filter were proposed in the literature. Only partial solutions are given, for the partial stability proof under many assumptions. In this study, a method based on Lyapunov approach, providing the stability region with all the necessary and sufficient conditions of the SDRE non-linear filter stability, is developed.

- Proof of Stability

From Equations (3.26) and (3.27), we can write:

$$\dot{\hat{x}} = F(\hat{x})\hat{x} + PH^T(\hat{x})R^{-1}[y(x) - H(\hat{x})\hat{x}] \quad \textbf{(3. 59)}$$

$$\dot{\hat{x}} = (F(\hat{x}) - PH^T(\hat{x})R^{-1}H(\hat{x}))\hat{x} + PH^T(\hat{x})R^{-1}y(x) \quad \textbf{(3. 60)}$$

By proposing a definite positive Lyapunov function $V(\hat{x}) = \hat{x}^T P^{-1}\hat{x} > 0$, the SDRE filter stability is guaranteed iff $\dot{V}(\hat{x}) < 0$.

Using the proposed Lyapunov function $\dot{V}(\hat{x}) = \dot{\hat{x}}^T P^{-1}\hat{x} + \hat{x}^T P^{-1}\dot{\hat{x}}$. As $\dot{P}^{-1} = 0$ (steady-state problem), then:

$$\dot{V}(\hat{x}) = 2\dot{\hat{x}}^T P^{-1}\hat{x} \quad \textbf{(3. 61)}$$

Replacing (3.60) in (3.61), we obtain:

$$\dot{V}(\hat{x}) = 2[\hat{x}^T F^T(\hat{x})P^{-1}\hat{x} + \hat{x}^T H^T(\hat{x})R^{-1}(y(x) - H(\hat{x})\hat{x}] \quad \textbf{(3. 62)}$$

Thus, if $\hat{x}^T F^T(\hat{x})P^{-1}\hat{x} < 0$ and $\hat{x}^T H^T(\hat{x})R^{-1}[y(x) - H(\hat{x})\hat{x}] < 0$ Then $\dot{V}(\hat{x}) < 0$

Let us start proving the first term is defined negative

1) $\hat{x}^T F^T(\hat{x})P^{-1}\hat{x} < 0$ **(3. 63)**

This inequality is equivalent to have $F(\hat{x})P^{-1}$ is definite negative. However $P^{-1} > 0$ and $F(\hat{x})$ is the state dependent coefficient form of $f(x)$.

Proof

As mentioned earlier in the paper, the SDC forms of a nonlinear function f are not unique. In the following we propose to derive $F_1(\hat{x})$ and $F_2(\hat{x})$ as two SDC forms of f. Then $F(\hat{x},\alpha) = \alpha F_1(\hat{x}) + (1-\alpha)F_2(\hat{x}),\ \alpha \in \Re$ is also an SDC form of f as

$$[\alpha F_1(\hat{x}) + (1-\alpha)F_2(\hat{x})]\hat{x} = \alpha F_1(\hat{x})\hat{x} + (1-\alpha)F_2(\hat{x})\,\hat{x} = \alpha f(\hat{x}) + (1-\alpha)\,f(\hat{x}) = f(\hat{x})$$

$F(\hat{x},\alpha)$ represents an infinite family of SDC parameterisations. We will adopt this combined SDC form of f for the development of our proof. α is an extra degree of freedom that can be used to build a suitable SDC form. From an infinite SDC form candidates, obeying to the three first following statements.

if $F_1(\hat{x}) < 0$ *and* $F_2(\hat{x}) < 0$ *then if* $0 < \alpha < 1 \Rightarrow F(\hat{x}) < 0$
if $F_1(\hat{x}) < 0$ *and* $F_2(\hat{x}) > 0$ *then if* $\alpha > 1 \Rightarrow F(\hat{x}) < 0$
if $F_2(\hat{x}) > 0$ *and* $F_2(\hat{x}) < 0$ *then if* $\alpha < 0 \Rightarrow F(\hat{x}) < 0$
if $F_1(\hat{x}) > 0$ *and* $F_2(\hat{x}) > 0$ *then if* $0 < \alpha < 1 \Rightarrow F(\hat{x}) > 0$

Thus, a good choice of α with possible SDC forms $F_1(\hat{x})$ and $F_2(\hat{x})$ assures (3.63) to be definite negative. Worth to mention that for a more general case of $k+1$ distinct SDC parameterisations, the dimension of α will be of order *k* and $F(\hat{x},\alpha)$ will have the following form:

$$F(x,\alpha) = (1-\alpha_k)F_{k+1}(x) + \sum_{i=1}^{k}(\prod_{j=1}^{k}\alpha_j)(1-\alpha_{i-1})F_i(x) \qquad \textbf{(3. 64)}$$

Let us examine, now, the second term, in (3.62), to prove its definite negative property:

2) $\hat{x}^T H^T(\hat{x}) R^{-1}[y(x) - H(\hat{x})\hat{x}] < 0$ **(3. 65)**

Assuming that the covariance matrix R is diagonal $R = diag(\sigma^2 x, \sigma^2 y, \sigma^2 z)$ and the observation is the GPS signal $y(x) = [x_{gps} \quad y_{gps} \quad z_{gps}]^T$, which is used by the observation matrix given by Equation (2.10), the Inequality (3.65) becomes:

$$\frac{1}{\sigma^2 x}\hat{x}(x_{gps} - \hat{x}) + \frac{1}{\sigma^2 y}\hat{y}(y_{gps} - \hat{y}) + \frac{1}{\sigma^2 z}\hat{z}(z_{gps} - \hat{z}) < 0 \qquad \textbf{(3. 66)}$$

$$\frac{(\hat{x}-\frac{x_{gps}}{2})^2}{\sigma^2 x}+\frac{(\hat{y}-\frac{y_{gps}}{2})^2}{\sigma^2 y}+\frac{(\hat{z}-\frac{z_{gps}}{2})^2}{\sigma^2 z}>D \qquad \textbf{(3. 67)}$$

With $D=(\frac{x_{gps}}{2\sigma x})^2+(\frac{y_{gps}}{2\sigma y})^2+(\frac{z_{gps}}{2\sigma z})^2$

This inequality defines a region outside an ellipsoid, which centre is $C=[\frac{x_{gps}}{2}\quad\frac{y_{gps}}{2}\quad\frac{z_{gps}}{2}]^T$ and semi-majors are: $r_x=\sigma x.\sqrt{D}$, $r_y=\sigma y.\sqrt{D}$, $r_z=\sigma z.\sqrt{D}$.

As long as a right choice of the parameter α assures the definite negative property of the first term of $\dot{V}$, this region defines the stability region, as shown in Figure (3.2); of our SDRE non-linear filter.

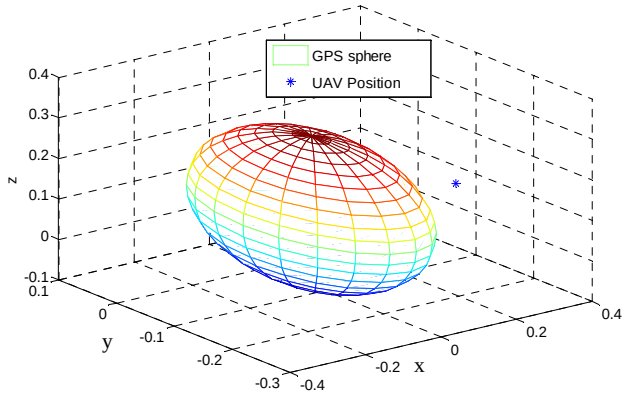


Figure.3. 2 Ellipsoid region of stability

Let us now discuss the stability of the SDRE nonlinear filter if the states of the systems go inside the ellipse, which implies that we fall in the case of $\dot{V}\geq 0$. In this case and from Equation (3.61), we get:

$$\dot{\hat{x}}^T P^{-1}\hat{x}=\hat{x}^T P^{-1}\dot{\hat{x}}\geq 0 \qquad \textbf{(3. 68)}$$

Then, for small values of Δt, we can use numerical derivative to obtain:

$$\hat{x}_k{}^T P^{-1}(\frac{\hat{x}_{k+1}-\hat{x}_k}{\Delta t})\geq 0 \qquad \textbf{(3. 69)}$$

$$\hat{x}_k{}^T P^{-1}(\hat{x}_{k+1}-\hat{x}_k)\geq 0 \qquad \textbf{(3. 70)}$$

$$\hat{x}_k{}^T P^{-1} \hat{x}_{k+1} \geq \hat{x}_k{}^T P^{-1} \hat{x}_k \quad \textbf{(3. 71)}$$

To well understand the meaning of these equations let us represent the Inequality (3.71) by vectors: $\vec{i}$ of coordinates $\hat{x}_k{}^T P^{-1}$, $\vec{e}_k$ of coordinates $\hat{x}_k$ and $\vec{e}_{k+1}$ of coordinates $\hat{x}_{k+1}$. Then Inequality (3.71) becomes:

$$\vec{i} \cdot \vec{e}_{k+1} \geq \vec{i} \cdot \vec{e}_k \quad \textbf{(3. 72)}$$

On the other hand and because $P^{-1} > 0$, $\hat{x}_k{}^T P^{-1} \hat{x}_k > 0$, which means:

$$\vec{i} \cdot \vec{e}_k > 0 \quad \textbf{(3. 73)}$$

This inequality signifies that $\cos(\vec{i}, \vec{e}_k) > 0$, which means that the angle between $\vec{i}$ and $\vec{e}_k$, is stuck between $]-\pi/2, \pi/2[$ as shown in Figure (3.3). Thus, we can state that the vector $\vec{i}$ diverges from the origin and tries to go out of the ellipse. Combining this result with the Inequality (3.72), we can conclude that the projection of $\vec{e}_{k+1}$ on $\vec{i}$ is larger than the projection of $\vec{e}_k$ on $\vec{i}$, which implies that the vector $\vec{e}_{k+1}$ diverges from the origin more than $\vec{e}_k$. We obtain, then, the following final result:

$\forall \vec{e}_k\ (\hat{x}_k)$, such that $\|\vec{e}_k\| < D$, then $\vec{e}_{k+1}(\hat{x}_{k+1})$ attempts to diverge from the origin, i.e. tries to return back (attraction) to the stability region.

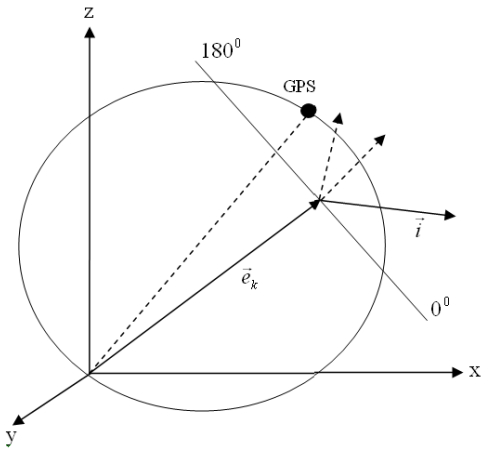


Figure.3. 3 Stability inside the ellipse

This result shows that, whenever the SDRE filter provides state estimates bringing the system at the frontiers of the inside ellipsoid region, which implies $\dot{V} \geq 0$ and thus no

guarantees of the global filter stability, we showed that the SDRE nonlinear filter will bring back the system to the region of stability shown on Figure (3.2).

The theoretical results of the stability study undertaken above give us safe flight regions for the UAV in terms of robust navigation. We did proof that the exterior of the ellipsoid Figure (3.2) is a stability region and every time our UAV is in this region it will be attracted to the GPS ellipsoid as shown in Figure (3.4). In Figures (3.4.a, b, c, d, e and f) the UAV is attracted to the GPS ellipsoid, then in Figure (3.4.g) the UAV position becomes inside the ellipsoid, before going out again in Figures (3.4.h and e) . The convergence of the GPS (red) and SDRE (blue) radiuses is given by Figure (3.5).

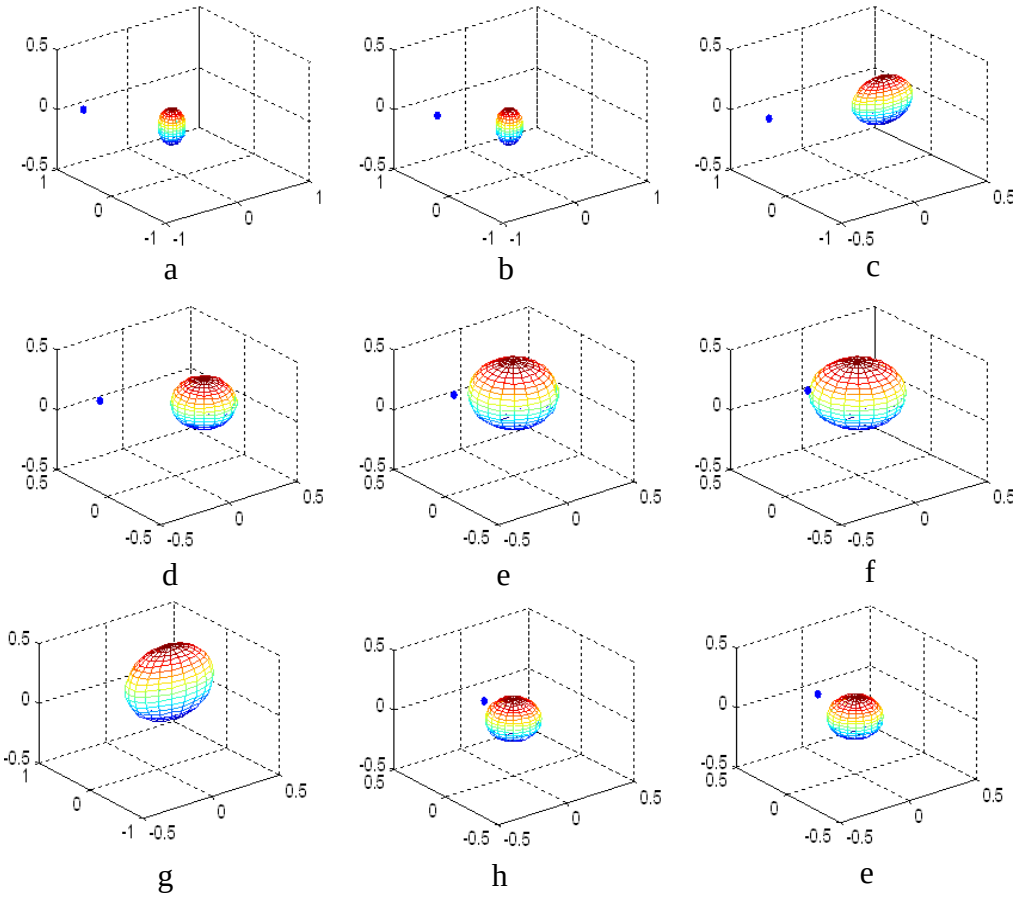


Figure.3. 4 UAV position and Stability sphere in the navigation frame

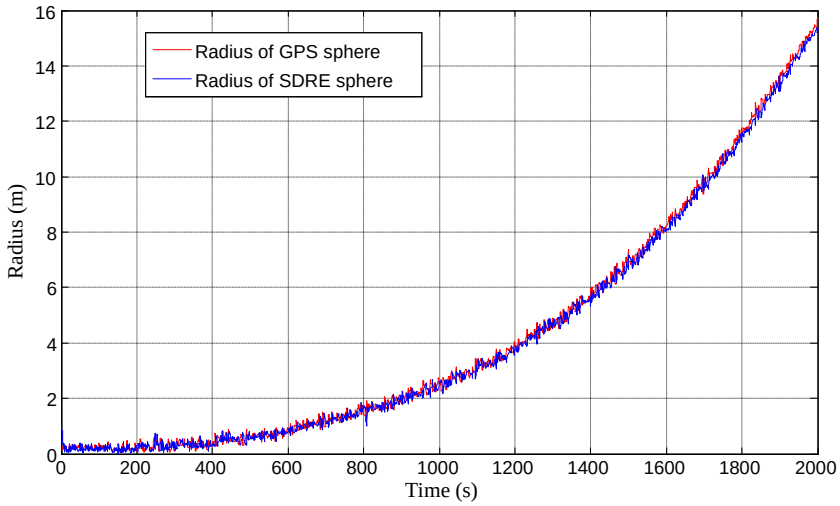


Figure.3. 5 GPS and SDRE Sphere radius during the time of simulation

3.7.2. Simulation results

In fact, simulation results of Kalman Filter applied on INS error model (small Euler angles), Extended Kalman Filter based on the model linearisation, Unscented Kalman Filter (UKF), the proposed SDRE non-linear filter and the nonlinear $H\infty$ filter will be presented as alternative techniques to estimate the state of the nonlinear system.

The sampling rates used for each sensor and filter adopted in this study are as follows, $f_{INS} = 100Hz$, $f_{GPS} = 1Hz$, $f_{EKF} = 10Hz$, $f_{SDRE} = 10Hz$, $f_{NH\infty} = 10Hz$. Every alternate 0.1s instant, when the GPS update is not available, Equations (3.8, 3.21 and 3.26) are used (respectively With KF, EKF and SDRE) to update the current state $\hat{x}_k$ using the most recent GPS update as the measurement instead of the predicted state; this choice will not affect the performance of our filter because of the slight position variation.

3.7.2.1 SDRE vs KF, EKF and UKF

Simulation results shown in Figures (3.6, 3.7 and 3.8) represent the estimated airborne position given by the KF, EKF and SDRE filter respectively, for the same 3D trajectory given by Figure (3.11).

The KF estimation is adequate only for small airborne angular rate, however if this condition doesn't hold then the estimation performances may be poor and the localisation error is significant as presented in Figure (3.6.a, b, c).

Figure (3.7.a, b and c) shows that the EKF provides good estimation for smooth trajectory Figure (3.7.a, b). Unfortunately, EKF performance is poor against heavy non-linearity cases where the Jacobian matrix is ill conditioned causing undesirable peaks in the estimated coordinate as shown in Figure (3.7.c). Figure (3.8) shows the results obtained by the proposed SDRE non-linear filter based on two SDC forms F_1 and F_2 with a suitable value of the parameter $\alpha = 0.5$. It is clear that the estimation results are improved with the SDRE filter. In addition, we can observe that for strong non-linearity, the estimation error increase slightly but it still tolerable for the localisation problem comparing with the others methods.

Figure (3.9) presents a comparison between Unscented Kalman Filter (UKF) and the non-linear SDRE filter. As can be seen from, both UKF and SDRE provide good and similar estimations of the UAV z position. This similarity in performance between SDRE and UKF generally holds. However, UKF filter as expected and as mentioned in the introduction of this chapter is computationally heavier (Table (3.2)), because of the unscented transformation applied on each sigma-point. Table (3.1) shows comparison (after many tests) of the standard deviation between the true output (state) and the filter output for the implemented filters KF, EKF, UKF and SDRE non-linear filter, to show overhaul filters performances. This table confirm the previous results.

Finally, GPS, INS, true and estimated (EKF, UKF and SDRE) 3D UAV trajectories are shown in Figure (3.10). From this latter we can definitively notice that SDRE estimated trajectory is smoother than the UKF estimated trajectory. This is expected as we are dealing with a highly nonlinear navigation system and a sharp UAV trajectory.

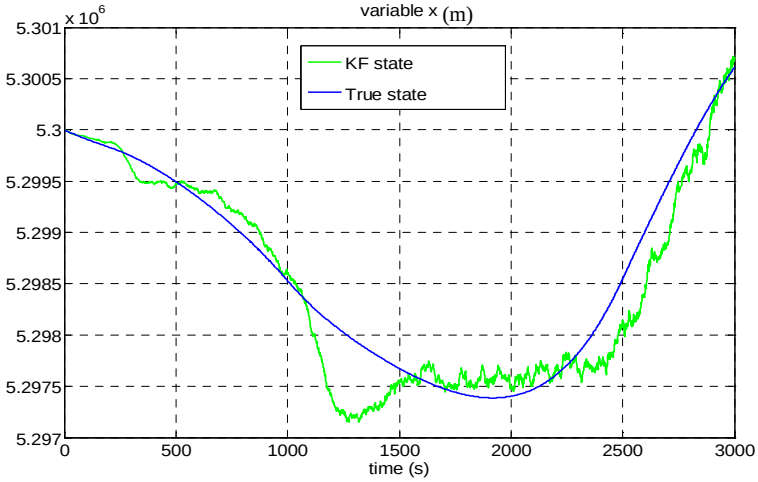
	$\sigma_x(m)$	$\sigma_y(m)$	$\sigma_z(m)$
KF	59.0247	23.8104	49.2331
EKF	3.4187	2.9710	7.8191
UKF	3.1234	3.1995	3.5681
SDRE	4.0513	1.2580	3.0705

Table 3. 2 Comparison of the standard deviation between KF (linear error model), EKF (linearized model) and SDRE (Non-linear system)

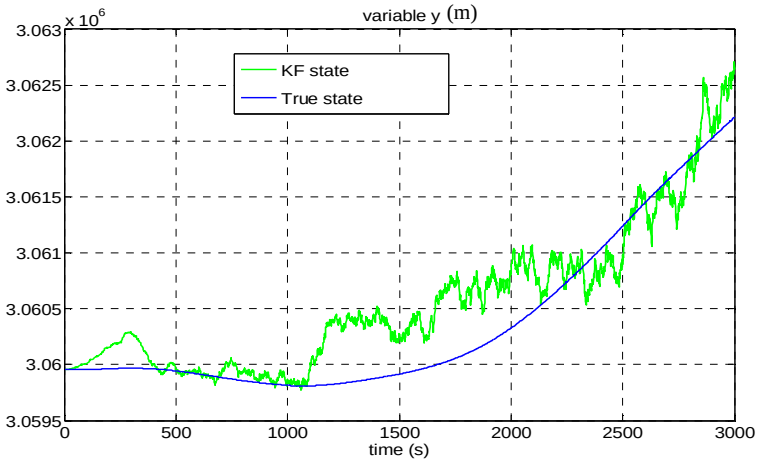
	EKF	UKF	SDRE
Required time for 100 iterations (s)	0.5670	8.4500	1.5000

Table 3. 3 Comparison of the computation time between

A. Kalman Filter Applied to a Linear INS model

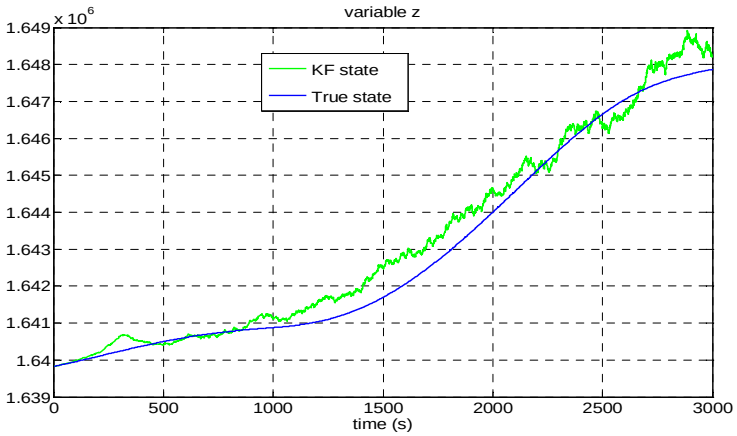


-a-



-b-

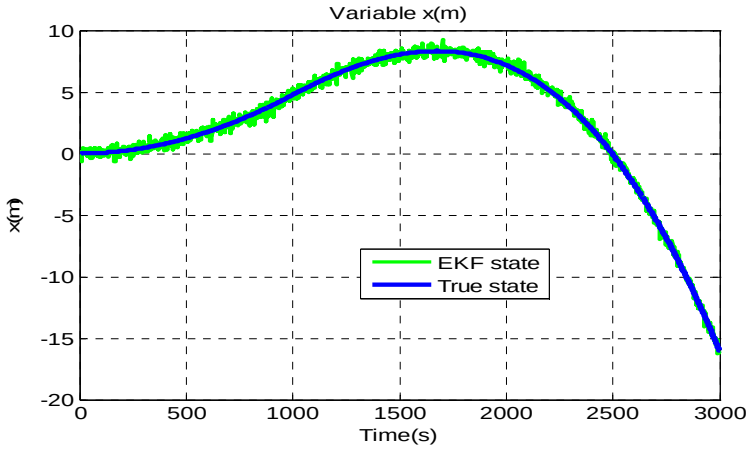
(m)



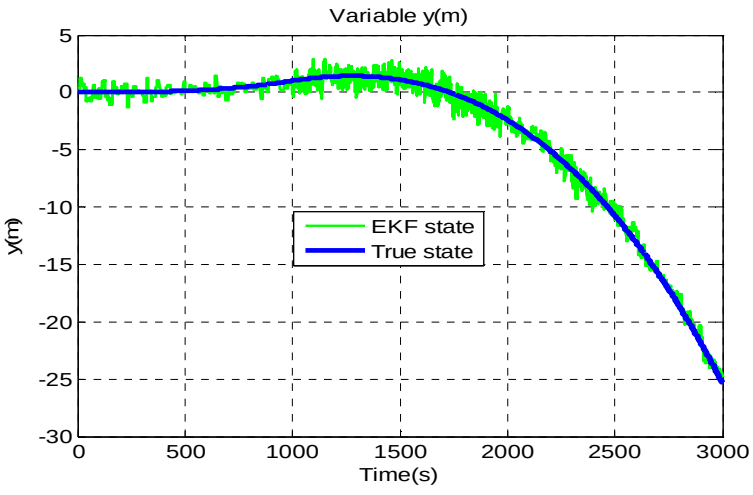
-c-

Figure.3. 6 Position of the Airborne, with Kalman Filter *(Linear error model)*

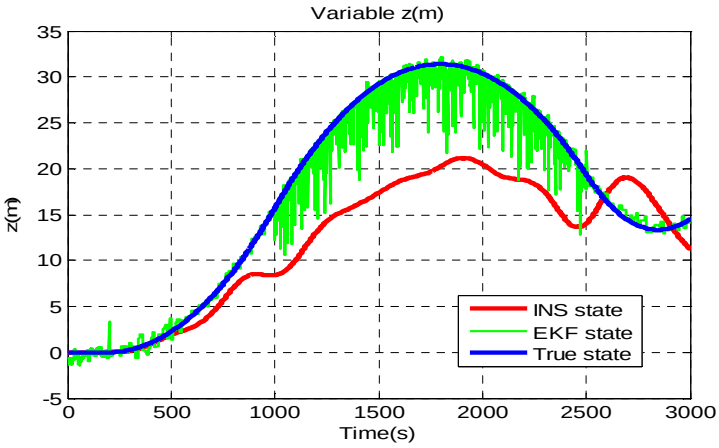
B. Extended Kalman Filter Applied to a Linearised INS model



-a-



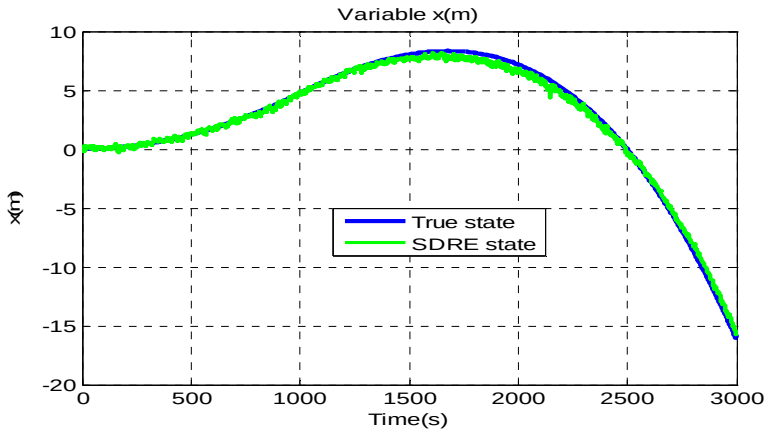
-b-



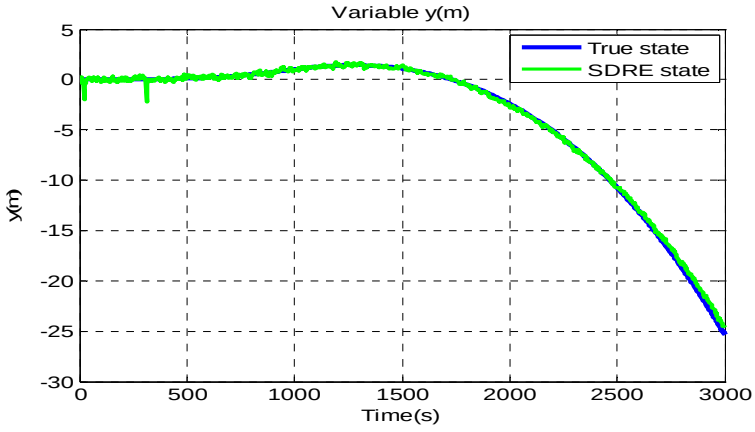
-c-

Figure.3. 7 Airborne position, by the Extended Kalman Filter

C. SDRE Nonlinear Filter Applied on the Nonlinear INS model



-a-



-b-

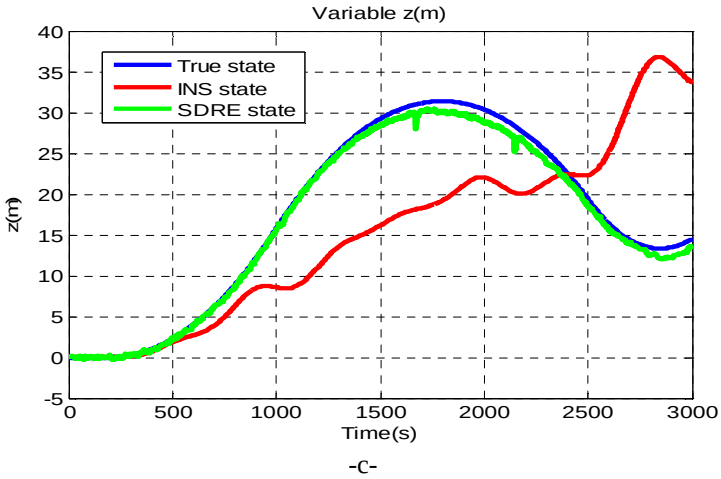


Figure.3. 8 Position of the Airborne, with SDRE Nonlinear filter (Non-linear model)

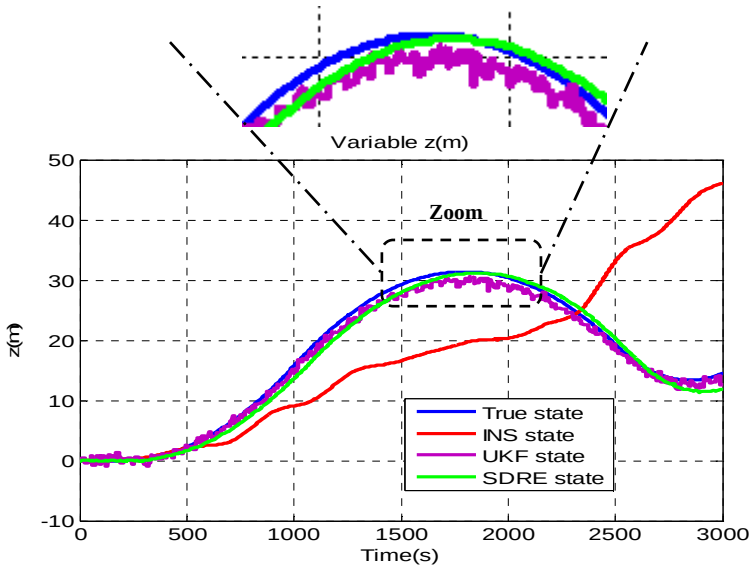


Figure.3. 9 Estimated of the airborne z position, comparison between UKF and SDRE

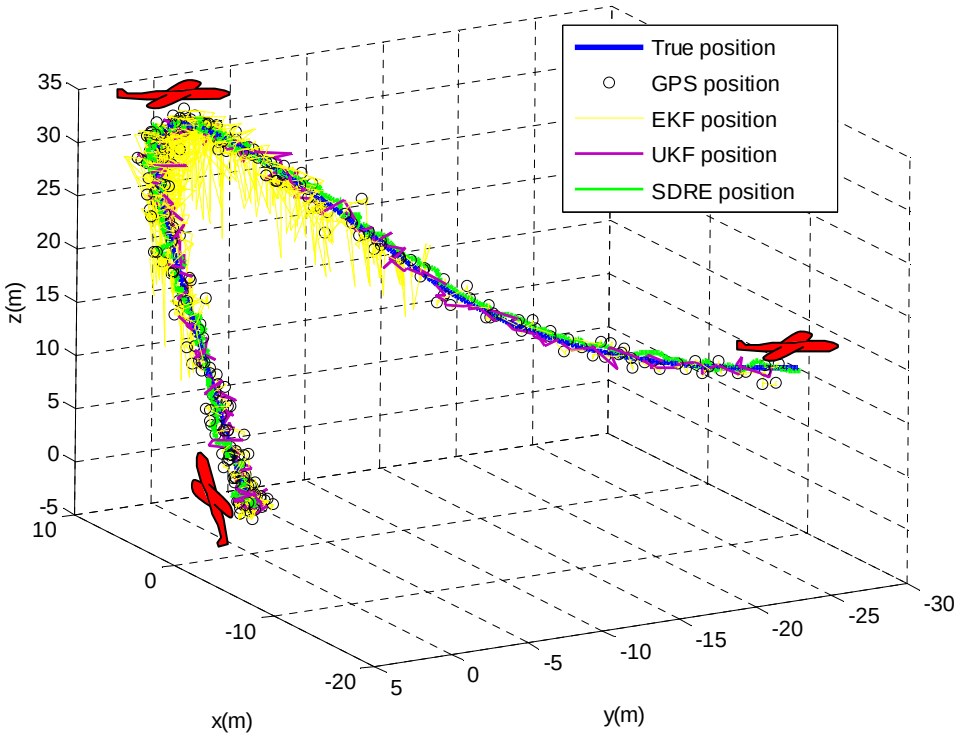
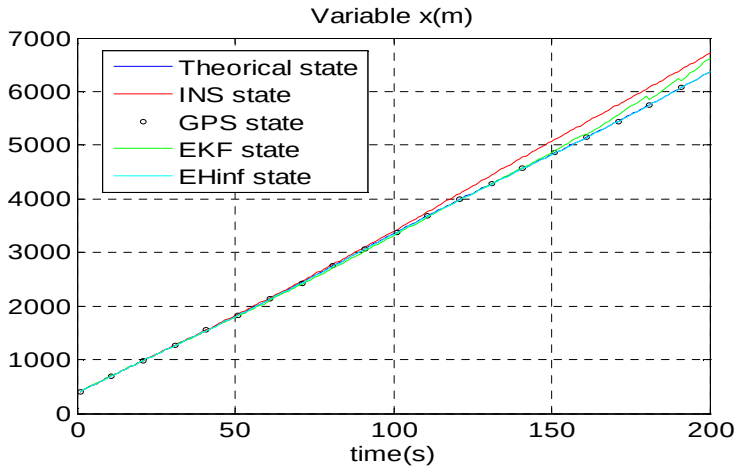


Figure.3. 10 UAV trajectory and estimated positions

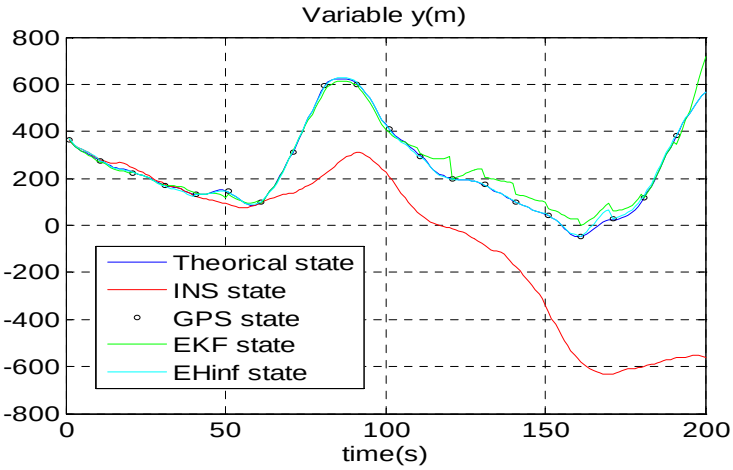
3.7.2.2 NH∞ vs EKF

A. Position (x, y, z)

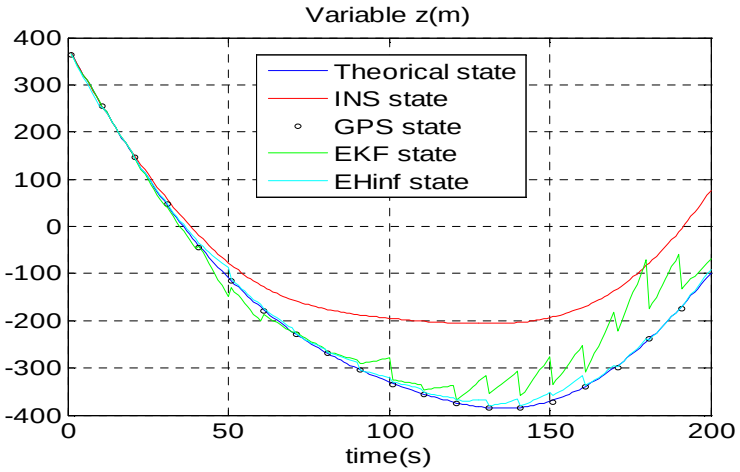
Figure (3.11.a, b, and c) shows the position estimation of the UAV following the axis (x, y and z) respectively, obtained by the Nonlinear H∞ filter, as can be seen from these figures a good performances are obtained by this filter comparing to the previous filters (KF, EKF,…). Moreover, the H∞ nonlinear filter does not need any assumption on the system or the noises affecting this system, which makes our filter robust against any type of disturbances. However, in this example, between two GPS observations, and when the GPS update is not available, Equation (3.46) is used to predict the state $\hat{x}_k$, which improves significantly the localisation accuracy.



-a-



-b-

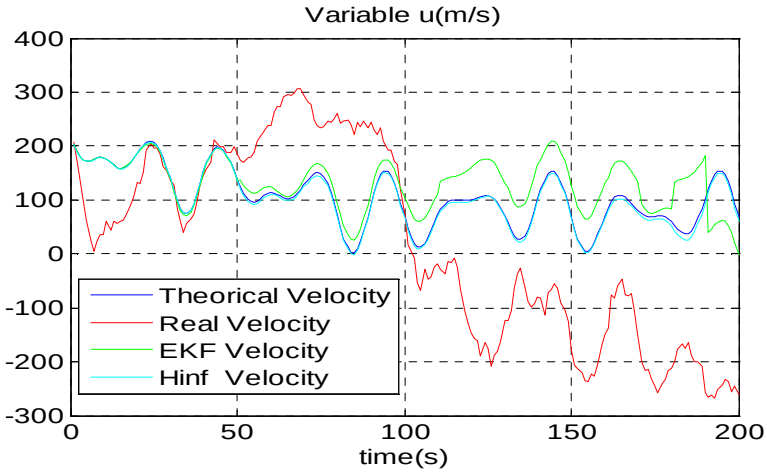


-c-

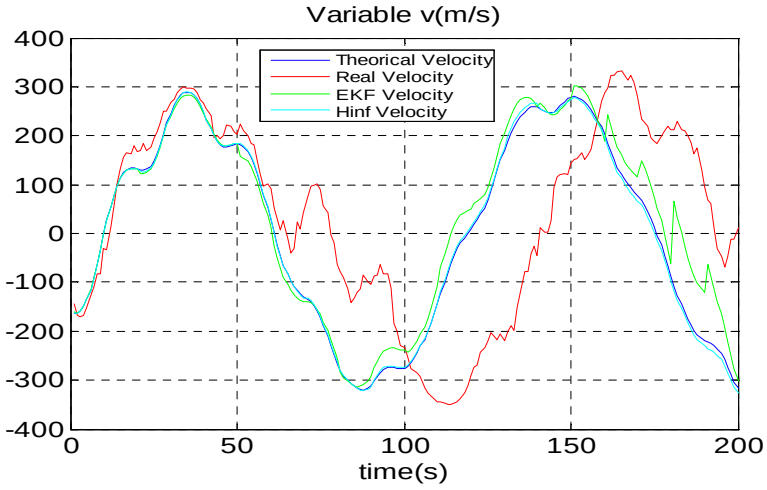
Figure.3. 11 Estimation of the UAV position

B. Velocity (u, v, w)

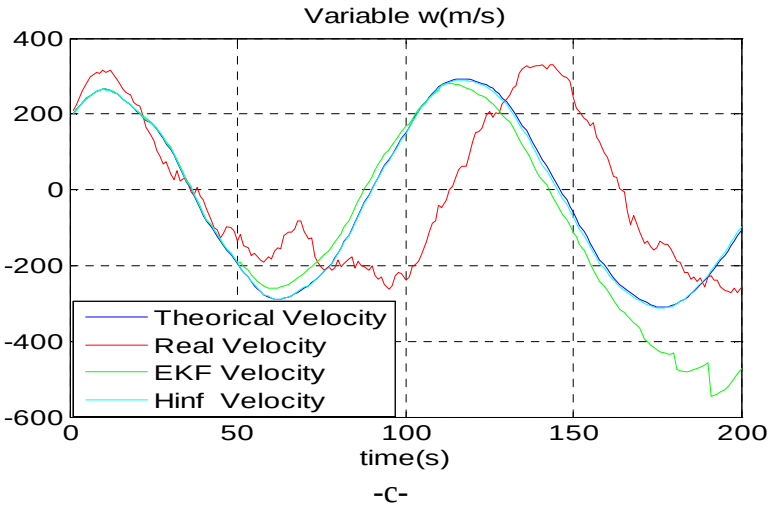
The Figure (3.12. a, b and c) show the evolution of the UAV velocities (u, v and w) following the three axes (North, East, Down) respectively. As can be seen from these figures the nonlinear $H\infty$ filter performs better than the EKF and both filters perform much better than the INS velocities.



-a-



-b-

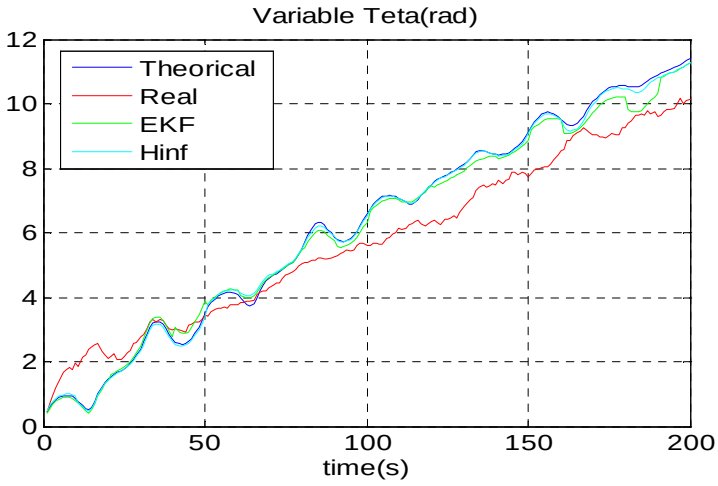


-c-

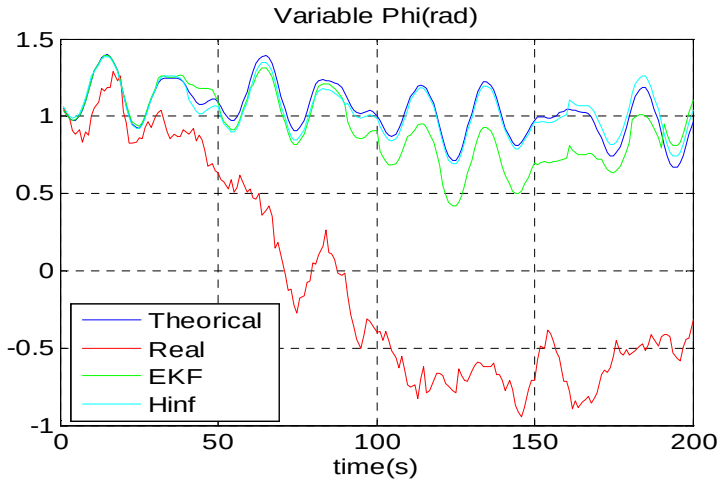
Figure.3. 12 Estimation of the UAV velocity

C. Euler Angle (θ, ϕ, ψ)

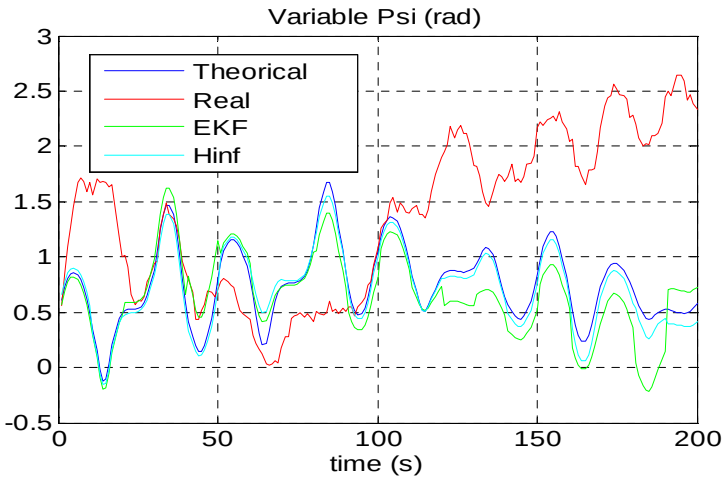
The same observations leading to the conclusions about the velocity estimation can be observed for the Euler angles in Figure (3.13.a, b and c).



-a-



-b-



-c-

Figure.3. 13 Estimation of the UAV Euler angle

Figure (3.14) shows the UAV 3D trajectories given by the INS (red), EKF (green), NH∞ filter (cyan), GPS (circle). These trajectories are compared with the theoretical one. As shown, on the figure, the chosen scenario presents a heavy nonlinearity allowing us to show the drawback of a bad linearisation. It is clear that the trajectory given by the NH∞ filter is more accurate than that given by the EKF.

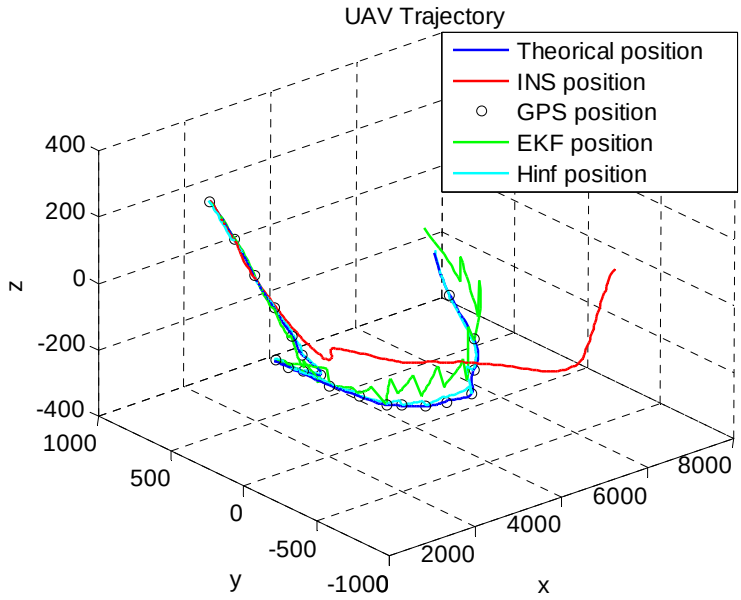


Figure.3. 14 UAV localisation

3.8 Conclusion

In this chapter, INS/GPS UAV localisation problem is investigated. Conventional and innovative filtering approaches are implemented and validated. The proposed SDRE nonlinear filter allows us to solve issues related to linearisation, which poses problem for the classical filtering techniques like EKF. Good results were obtained with the SDRE nonlinear filter comparing to the KF, EKF and UKF particularly in the case of strong nonlinearities. Formal proofs of the SDRE nonlinear navigation filter stability were proposed and a stability attractive region was determined.

Robust filters are also explored in this chapter. $H\infty$ and Nonlinear $H\infty$ are very promising filters first because their robustness and second because they doesn't make any assumption about noises characteristics.

Chapter 4

UAV map building

CHAPTER 4

UAV map building

4.1 Overview

The following chapter is divided in three parts. In the first part, some feature detectors and descriptors are presented. An adaptive version of the Scale Invariant Feature Transform (SIFT) is proposed and then is followed by a feature stability analysis. In the second part, an improved image mosaicing algorithm using superresolution is proposed using Adaptive SIFT features. Finally, in the third part of the chapter a new robust and fast feature detector and descriptor called Scale Wavelet Feature Transform (SWIF) is proposed. The SWIF algorithm is based on the Multi wavelet decomposition and very suitable for VSLAM problem.

4.2 Part I: Feature Extraction & Matching for VSLAM

4.2.1 Introduction

A critical issue in a SLAM algorithm is to decide how to represent the joint distribution over vehicle poses and maps. In particular, the map representation is crucial due to the fact that maps are usually represented by a high number of parameters. The two mostly used map representations are the landmark based maps and the dense maps (also called grid maps) [69]. In this work, we are interested only in the landmark based maps.

4.2.2 Landmark Detection

The landmark maps represent the world as a set of spatially located features. When using this kind of representation, the main advantage is that the representation is compact, and therefore suitable for operating in large environments. Moreover, given that the map can be represented as a point in $\Re^{l.n}$ where l is the dimension of one

landmark and *n* is the number of landmarks, it becomes possible to represent the posterior over the whole map.

Incremental building and maintaining of maps for immediate use by a navigating airborne has been shown to rely on detailed knowledge of the cross-coupling between running estimates of the vehicle locations and mapped features [70]. Without this information, features which are redetected after a period of neglect are treated as new. The entire structure suffers from progressive error accumulation which depends on the distance travelled and not on distance from the starting position in the fiducial coordinate frame [71]. It becomes impossible to build persistent maps for long-term use as earlier navigation research shows [72-77].

In this section, we study the most important proposed approaches to detect invariant and distinctive features from an image. Also we propose some alternatives to improve the robustness of the detectors and to make them suitable to solve the Visual Airborne SLAM.

4.2.3 Image Feature Detectors

Many feature detectors are proposed in the literature. In the following paragraphs the most used detectors are reviewed.

4.2.3.1 Harris Corner Detector

The Harris corner detector is a popular interest point detector due to its strong invariance to rotation, illumination variation and image noise [78]. The Harris corner detector is based on the local auto-correlation function of a signal. This function measures the local changes of the signal with patches shifted by a small amount in different directions. A discrete predecessor of the Harris detector was presented by Moravec [79]; where the discreteness refers to the shifting of the patches. Given a shift $(\Delta x, \Delta y)$ and a point (x, y), the auto-correlation function is defined as:

$$c(x,y) = \sum_{w} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \qquad \textbf{(4. 1)}$$

where $I(\cdot, \cdot)$ denotes the image function and (x_i, y_i) are the points in the window W (Gaussian) centred on (x, y). The shifted image is approximated by a Taylor expansion truncated to the first order terms,

$$I(x_i + \Delta x,\ y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i)\ \ I_y(x_i, y_i)]\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} \quad \textbf{(4. 2)}$$

Where $I_x(\cdot,\ \cdot)$ and $I_y(\cdot,\ \cdot)$ denote the partial derivatives in x and y, respectively. Substituting approximation Equation (4.2) into Equation (4.1) yields,

$$\begin{aligned} c(x, y) &= [\Delta x\ \Delta y]\begin{bmatrix}\sum_w (I_x(x_i, y_i))^2 & \sum_w I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_w I_x(x_i, y_i) I_y(x_i, y_i) & \sum_w (I_y(x_i, y_i))^2\end{bmatrix}\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} \\ &= [\Delta x\ \Delta y] C(x, y)\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} \end{aligned} \quad \textbf{(4. 3)}$$

where, the $C(x,\ y)$ is the second moment matrix. It captures the intensity structure of the local neighbourhood. Let $\lambda 1, \lambda 2$ be the eigenvalues of matrix $C(x,\ y)$. The eigenvalues form a rotationally invariant description. There are three cases to be considered:

1. If both $\lambda 1, \lambda 2$ are small, so that the local auto-correlation function is flat (i.e. little change in $c(x,\ y)$ in any direction), the windowed image region is of approximately constant intensity.
2. If one Eigen-value is high and the other low, the local auto-correlation function is ridge shaped. Then only local shifts in one direction (along the ridge) cause little change in $c(x,\ y)$ and significant change in the orthogonal direction. This indicates an edge.
3. If both Eigen-values are high, so the local auto-correlation function is sharply peaked. Then shifts in any direction will result in a significant increase. This indicates a corner.

4.2.3.2 Harris-Laplace Detector

In [80], Mikolajczyk and Schmid propose an invariant scale method for detecting interest points. The first step of this method is to compute interest points (Harris points) at different scales. Then, points with a local maximal measure (the Laplacian) will be

selected as Harris-Laplace interest points. These interest points can be detected at different scales.

According to [80], local extrema over a scale of normalised derivatives indicates the presence of local structure characteristics. The scale characteristic can be found by searching for a local maximum over all scales. There are several derivative based functions (Laplacian, Difference-of-Gaussian and Harris function) that can compute a scale representation of a feature. Laplacian Equation (4.4) is used in Harris-Laplace interest point detection due to its high detection rate [80].

$$\left|\sigma^2 (L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma))\right| \quad \textbf{(4. 4)}$$

where L_{xx} and L_{yy} are the second image derivative following x and y respectively, and σ is the image scale.

Harris-Laplace interest points can be detected by comparing Laplacian at different scales. The scale of the point with a maximum Laplacian is taken as the characteristic scale of this interest point. The accuracy of the interest point detected is at a pixel level, which is not good enough for pose recovery. Parabola interpolation is used in this work to get the precise location of interest point to sub-pixel level. The output of Harris-Laplace detector are scale invariant points of different size. These points need to be described for indexing.

4.2.3.3 SIFT Features

Inspired by the response properties of complex neurons in visual cortex, Lowe proposed the Scale Invariant Feature Transformation algorithm (SIFT) [81]. SIFT features are distinctive and invariant from images that can be used to perform reliable matching between different images of the same object or scene. Firstly, it uses a scale-space extrema to efficiently detect the location of those stable keypoints in the scale and space. Then, an orientation histogram based on the gradient in different directions is formed around the keypoint and the dominant orientation is used to represent the keypoint orientations. Finally, a gradient histogram is created as a very distinctive descriptor of that keypoint. Thus, each keypoint is represented by the scale, orientation,

location and the gradient descriptor in a way that it can achieve scale and orientation invariance. Also, the descriptor is so distinctive that it can have high probability to find the exact match under certain extent of illumination changes and 6D affine transform. All these characteristics (especially the scale and orientation invariance) perfectly fulfill the basic requirements of the detection task. Following are the major stages of computation used to generate the set of SIFT features [81]:

1. Scale-space extrema detection

The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation, Figure (4.1.b).

2. Keypoint localisation

At each candidate location, a detailed model is fit to determine location and scale. Interest points are selected based on measures of their stability, for example features situated on edges are not stable and should be removed, Figure (4.1.c and d).

3. Orientation assignment

One or more orientations are assigned to each interest point location based on local image gradient directions. All future operations are performed on image data that have been transformed relative to the assigned orientation, scale, and location for each feature. Thereby, providing invariance to these transformations.

4. Feature descriptor

Local image gradients are measured at the selected scale in the region around each interest point. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

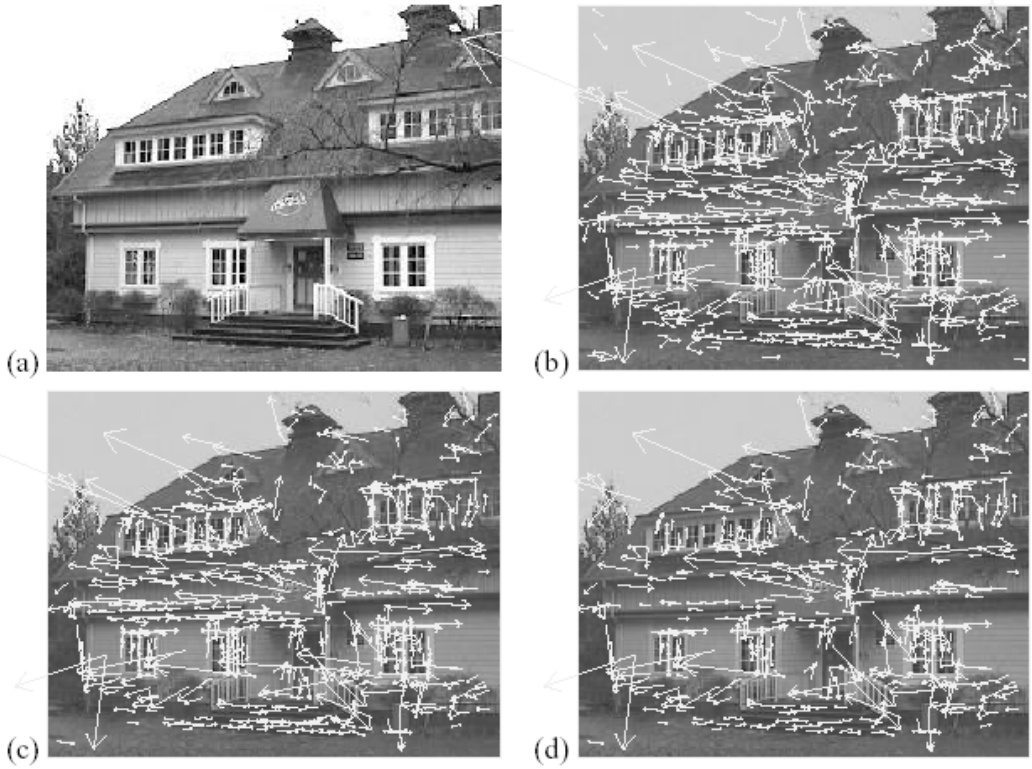


Figure.4. 1 Example of features localisation by the SIFT algorithm [81] (a) 233x189 image, (b) 832 DOG extrema, (c) 729 left after peak, value threshold (d) 536 left after testing ratio of principle curvatures

An important aspect of SIFT detector approach is that it generates large number of features that densely cover the image over the full range of scales and locations. A typical image of size 500x500 pixels will give rise to about 2000 features (although this number depends on both image content and choices for various parameters). The quantity of features is particularly important for object recognition, where few robust features are enough for robot localisation.

4.2.3.4 Limits of the SIFT

Despite the SIFT algorithm having many advantages, the detected keypoints are still not efficient, especially for Visual SLAM problem. SIFT algorithm uses a constant factor k between the scales. However, a big value of k implies few keypoints, which is a disadvantage for the VSLAM problem especially if some of these keypoints are unstable. Also, small values of k implies detection of lot of keypoints, which is a

problem for VSALM too. Clearly it is not evident to find the suitable value of k for all kind of images.

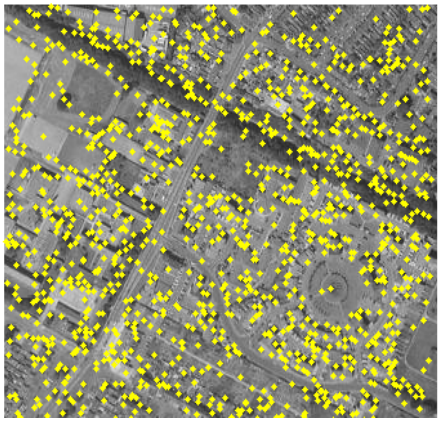
David Lowe proposes an adequate value of $k = 1.6$, which is efficient only for images with appropriate amount of texture. For example, for low frequency images (seas, deserts…) only few features are detected, Figure (4.2.d). On the other hand, for high frequency images (spatial, building…) SIFT algorithm leads to lots of features, which are not suitable for VSLAM algorithm, Figure (4.2.c). As a solution to this problem, an adaptive scale representation with an adaptive factor k is proposed.

4.2.4 Feature Detection with Adaptive Scale Factors *k*

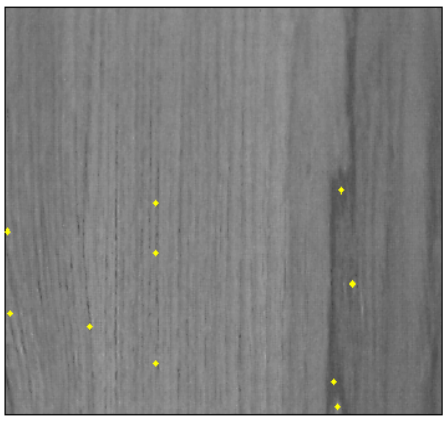
Figure (4.2.a, b), below, gives an example of two images, where Image 1 contains a lot of textures (high frequency image), and the second image, Image 2, contains only few textures (low frequency image). Figure (4.2.c, d) shows detected SIFT keypoints in the two images (Image 1 and Image 2).

a) Image1 (High frequency)

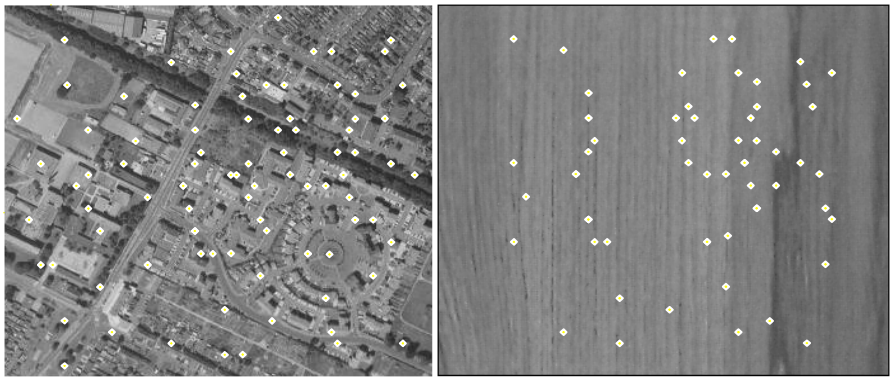
b) Image2 (Low frequency)



c) 1563 Keypoints



d) 12 Keypoints



e) 75 Keypoints f) 53 Keypoints

Figure.4. 2 Features detected by the SIFT algorithm with k=1.6

As can be seen from Figure (4.2.c), a large number (1563) of keypoints is detected. It is impossible to use such amount of keypoints by a VSLAM algorithm [8-11]. On the other hand, in Figure (4.2.d), only 12 keypoints are detected, which might be not enough for mapping especially if many of these keypoints are unstable. Thus, a way of determining an adaptive scale factor "k" is required. To find the suitable values of k, we propose to use the energy of the Difference of Gauss (DoG) image, which gives us an idea about the information contained in the image. The energy of an image is calculated as the sum of DoG squares:

$$E = \sum_{i=1}^{n} \sum_{j=1}^{m} D^2(i, j) \qquad \textbf{(4. 5)}$$

where $[n, m]$ is the size of the image, D is the DoG of the image. For a fixed value of k, Figure (4.3) presents calculated energies based on DoG for the image illustrated in Figure (4.6. a) at different scales. As can be seen from this figure the energy of the DoG image decreases exponentially when the scale increases. Figure (4.4) shows scale levels used with a fixed value of k for different smooth versions of the image. As shown in this figure, a large range of scales, up to $\sigma > 20$ is used. Also, From Figure (4.3) and Figure (4.4), we can observe that when the scale σ increases, the difference between image versions obtained by smoothing with two nearby scales (σ_1 and σ_2), decreases. Thus, no significant information is obtained by the smoothed image versions at high scales. As an

example, the figures below show that for a scale $\sigma = 20$, the energy of DoG of the image tends to zero. This means that the two image versions obtained by smoothing with σ and $k\sigma$ are almost similar (information redundancy).

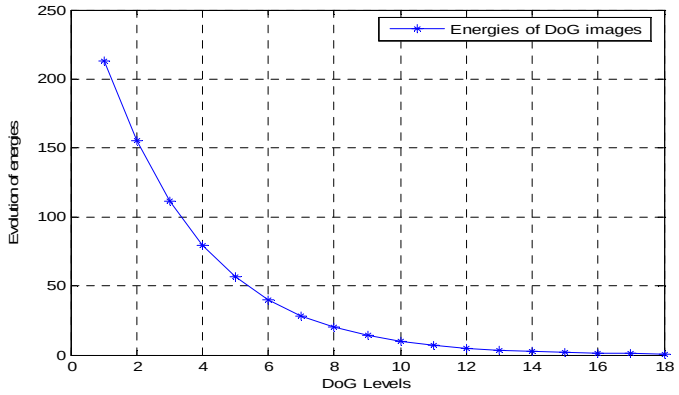


Figure.4. 3 Energies of the DoG images at different scales with *k=1.6*

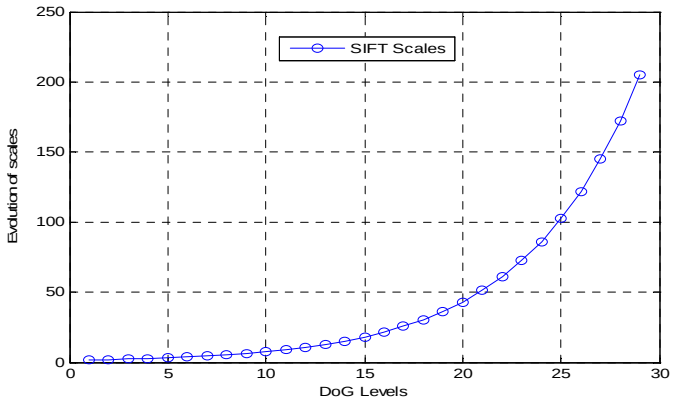


Figure.4. 4 Scales used, with a fixed *k*

4.2.5 Adaptive Scale Invariant Feature Transformation (ASIFT)

The proposed algorithm (Adaptive Scale Invariant Feature Transformation) is illustrated in Figure (4.5). The adaptive scale factor is adjusted following the energy of the DoG image. S_1 and S_2 are thresholds determined by experiments. The modification of k is calculated by the following equations:

$$\begin{cases} if \quad E < S_1 \quad \Rightarrow \quad k = k^2 \\ if \quad E > S_2 \quad \Rightarrow \quad k = \sqrt{k} \end{cases} \qquad \textbf{(4. 6)}$$

The proposed approach allows us to detect a suitable number of features while reducing the computation time. Figure (4.2. e and f) show features detected by the adaptive SIFT. As can be seen, a reasonable number of features are detected in both Images (high and low frequency).

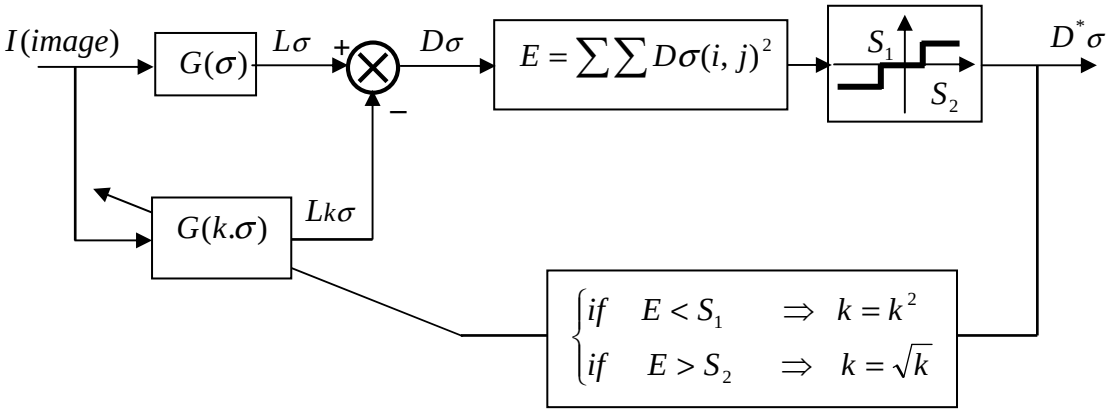
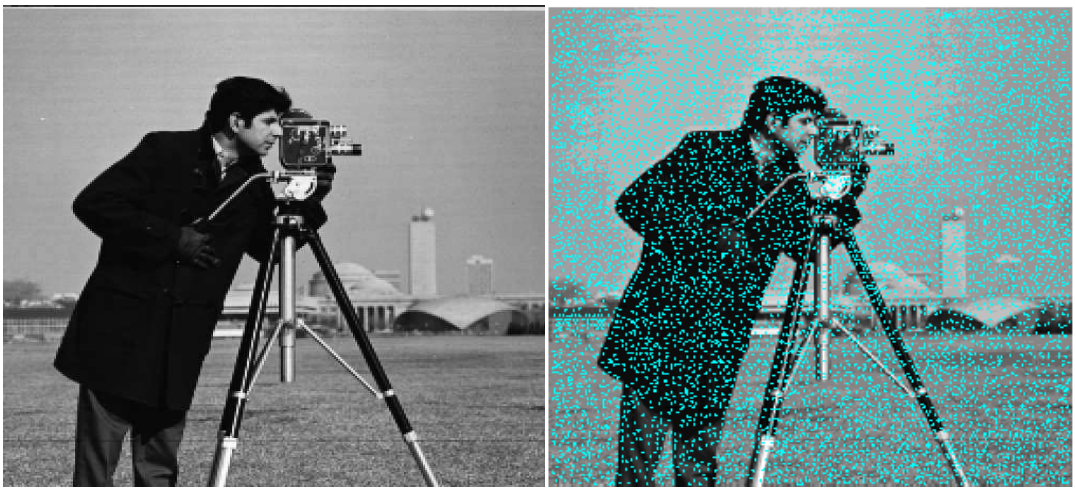


Figure.4. 5 Adaptive SIFT detector

The way we modify k helps to avoid the ineffective calculation of the DoG image. The proposed approach allows the detection of a suitable number of features. Also, it reduces the computation time. The following figures show feature localisation results by the proposed adaptive scale factor k. Figure (4.7.a), shows the detected features after removing false extrema. Then, unstable features, situated along the edges are removed as can be shown on Figure (4.7.b).



-a- -b-

Figure.4. 6 a) Original image, b) Extrema detection

-a- -b-

Figure.4. 7 Feature localisation a) keypoints localisation by an adaptive k b) Stable keypoints

Figure (4.8) and Figure (4.9) show the evolution of energy based on DoG and the scale evolution respectively for the image given in Figure (4.6.a), and with an energy threshold $E_s = 50$. As shown in Figure (4.8), when the image energy based on DoG decreases under the threshold E_s (dashed ellipse), a significant increase of scale appears in Figure (4.9) as a consequence. This scale change implies a rise of the image energy based on DoG as in Figure (4.8) (dashed ellipse). This process is repeated until the image energy based on DoG exceeds the threshold E_s or the scale exceeds a chosen value $\sigma_{\max} = 20$. Figure (4.10) shows the Difference of Gauss pyramid obtained by the adaptive SIFT algorithm. We observe that an obvious change appears from one level to another of the pyramid, which justifies the appropriate choices of k values.

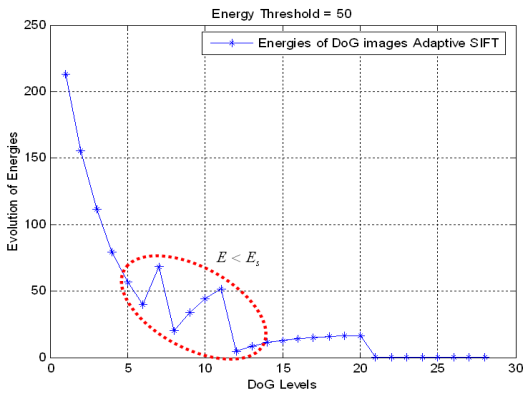


Figure.4. 7 Energies of the DoG images at different scales with an adaptive *k*

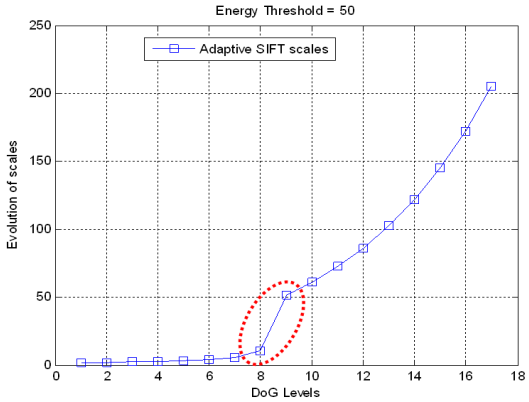


Figure.4. 8 Scales used, with an adaptive fixed *k*

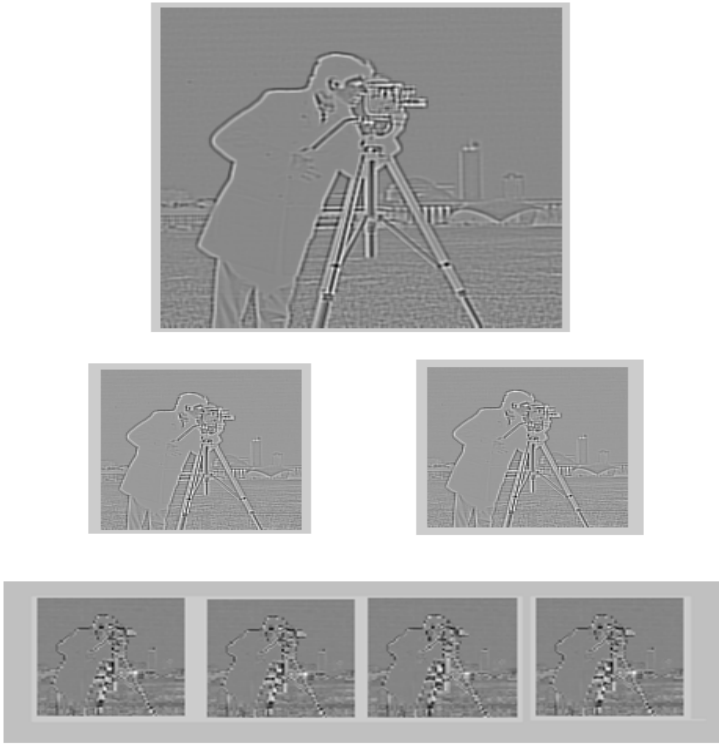


Figure.4. 9 Difference of Gauss (DoG) Pyramid

4.2.6 Feature Point Matching

A comparative study, in [80], of several local descriptors showed that the best matching results were obtained using SIFT algorithm [81]. However, as shown earlier, the original SIFT algorithm is not practical for Visual SLAM, especially, in environments of rich texture. Usually, from hundreds SIFT features extracted in one image, only 10% of these features can find their matches in another image from the

same sequence (successive images or stereo images) even when the change between the two viewpoints is small. In addition, many of the matches will be incorrect. Since map building algorithms should run in real time, extracting a large number of features, which can hardly find their matches, does not fulfil aerial vehicle speed requirements. Consequently, how to match SIFT features correctly is critical for vision-based SLAM.

Even if the adaptive SIFT, proposed in this chapter, allows the extraction of distinctive and invariant features, the proposed solution remains incomplete if no robust technique of matching is introduced.

Many criteria can be used to measure the distance between feature point descriptors such as orientation criteria and Euclidean distance (2-Norm). In addition to these distances, we propose in this paper to use Infinity norm distance ($\infty - norm$) for the feature matching problem based on SIFT descriptors. All the above-mentioned criteria are defined as follows:

Suppose p_i and p_j are two keypoints in image 1 and image 2, with descriptor D_i and D_j, respectively. Then, the distance between the two descriptors is given by:

- Orientation criteria:

$$Distance(i, j) = acos(\frac{D_i \cdot D_j}{\|D_i\| \cdot \|D_j\|}) = acos(D_i \cdot D_j)$$

- Euclidian distance:

$$Distance(i, j) = \|D_i - D_j\|_2 = \sqrt{(D_i - D_j)^T .(D_i - D_j)}$$

- Infinity norm:

$$Distance(i, j) = \|D_i - D_j\|_\infty = \max(|D_i - D_j|)$$

Then p_k is the correspondent of p_i if and only if

$$\begin{cases} Distance(i,k) = \min(Distance(i, j)) & j = 1 \ ... \ M \\ Distance(i,k) < DistRatio * \min(Distance(i, j)) & j = 1 ... M \ \ and \ \ j \neq k \end{cases}$$

Where M represents the keypoints number in the second image and *DistRatio* is a factor used to avoid the problem of similar descriptors.

The infinity norm criterion ensures more robustness; it takes into consideration the worst case (minimise the maximum of distances between the descriptor pairs). It is obvious then, that the matched features are as stable as possible. In the following, we present a comparative study for the matching problem using real images. These images were obtained by a stereo vision camera setup onboard an unmanned helicopter.

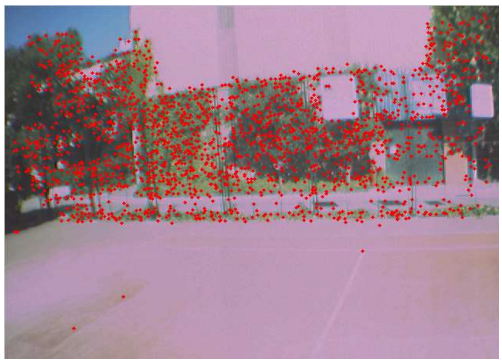


Figure.4. 10 Keypoints detected in Image 1 at time t1

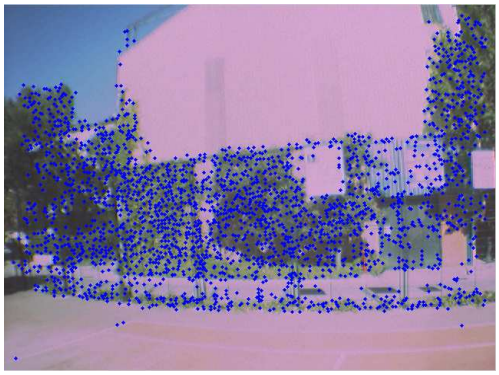


Figure.4. 11 Keypoints detected in Image 2 at time t2

A. Matching result

DistRatio = 0.6

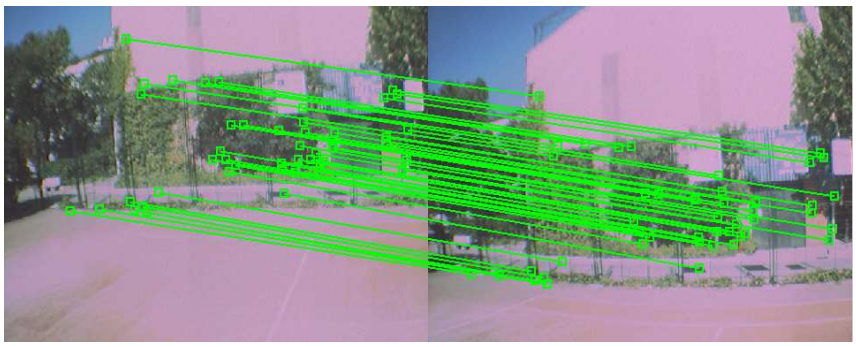


Figure.4. 12 with orientation distance, 52 good matching

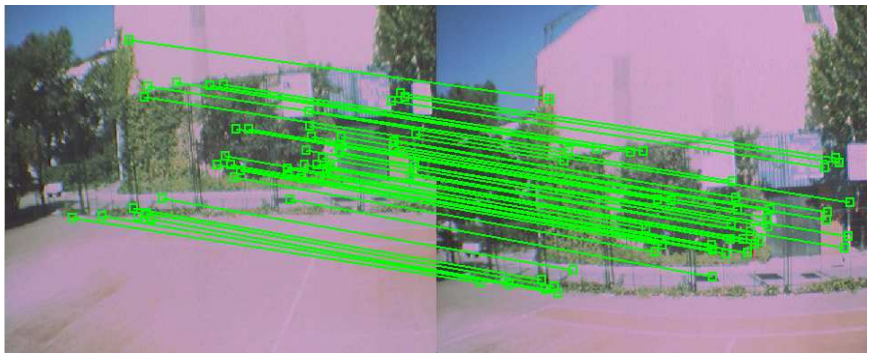


Figure.4. 13 with Euclidean distance, 51 good matching

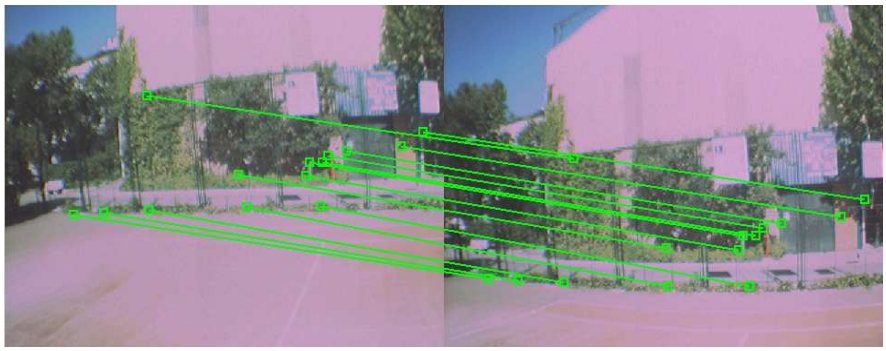


Figure.4. 14 with Infinity norm, 14 distinctive & good matching

As can be seen from Figure (4.15) the $\infty - norm$, allows the true association of a few but robust and distinctive features. This is very suitable for map building and Visual SLAM algorithm (intended application). Using the Euclidean and orientation distance Figures (4.13 and 4.14) many similar features are matched which increases the probability of false matching.

4.2.7 Feature Stability

Correct feature correspondence is not only linked with the matching criteria (previous section) but it is also linked with feature repeatability (unstable feature implies false match). To illustrate the problem of feature instability, an example showing the keypoints detected in the image of the same scene at different times is presented in Figure (4.16). At *t1* and at *t2* the red keypoints and the cyan keypoints were extracted respectively. Despite that SIFT feature extractor was applied on the same image, only few keypoints are redetected. This problem is even worst when acquired images are of different scenes.

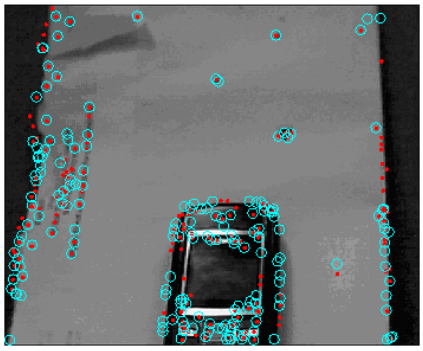


Figure.4. 15 Same image: red key points at time *t1*, cyan key points at time *t2*

SIFT algorithm is computationally heavy, especially during the calculation of DoG pyramids and more than that during the determination of the local extrema over all the scales and space. Many analyses showed that the most stable SIFT features are localised around one appropriate and true scale of the image. Mycalojski proposed, in [80], an algorithm for automatic scale selection based on the calculation of the Laplacian of Gaussians (LoG). However, it is time consuming. The following figures explain some analysis results that were achieved that were contribute to solve the feature stability problem.

Figure (4.17) presents an image of London seen from two different scales S_1 (left image) and S_2 (right image). The rate scale between the two images equals to ($S_1 / S_2 = 3$). Then, we use the SIFT algorithm to detect the keypoints in each image. We used the Euclidian distance (we could use the proposed Infinity norm as well) to match these keypoints based on their descriptor vectors. The histogram of the matched features

with respect to the scales of the two images, UK150%, UK50%, is given by Figure (4.18) and Figure (4.19) respectively.

Figure.4. 16 Image of UK given at two scales 150 %(left), 50 %(right)

From Figure (4.18), we observe that the most stable keypoints are detected at scale $\sigma \approx 3$, whereas from Figure (4.19), the most stable keypoints are detected at a scale $\sigma \approx 1$.

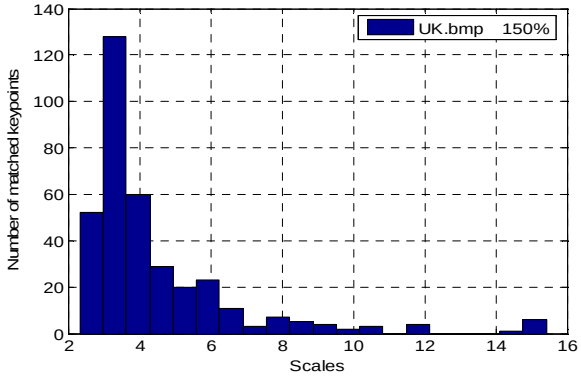


Figure.4. 17 Histogram of the keypoints scales (image left)

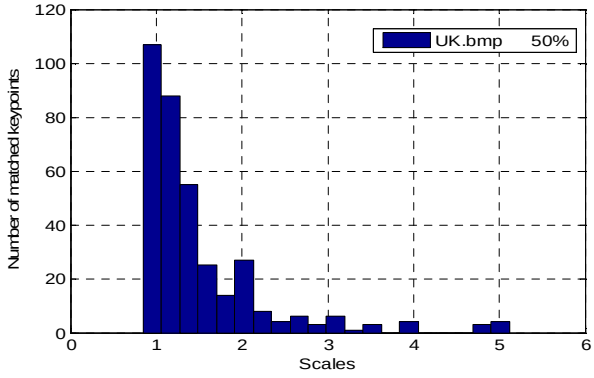


Figure.4. 18 Histogram of the keypoints scales (image right)

Furthermore, more than 90% of the matched keypoints (correct correspondence), as shown in Figure (4.20), are detected when the rate between the scales of the two matched keypoints is closely equal to $1/3$, which is the real rate scale between the two images. This result is illustrated and confirmed by Figure (4.21), which shows that the scale rates of the matched keypoints lie in the interval 3 ± 0.14.

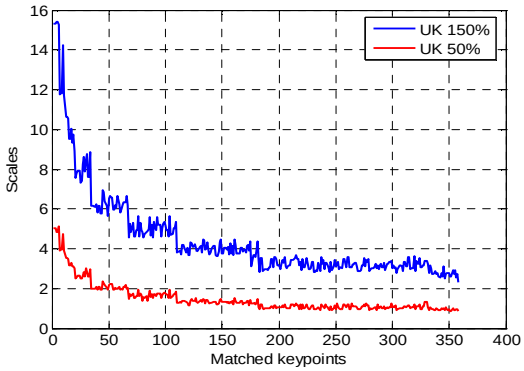


Figure.4. 19 Scales of the matched keypoints

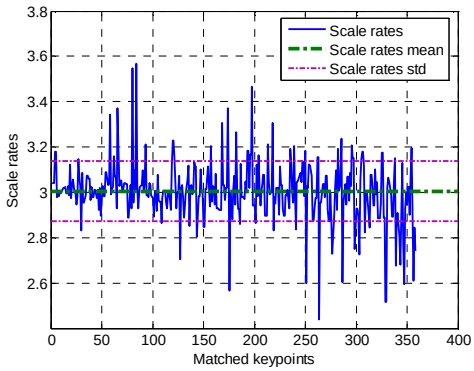


Figure.4. 20 Scale rate between the two images

4.2.8 ASIFT with Stereovision Constraints

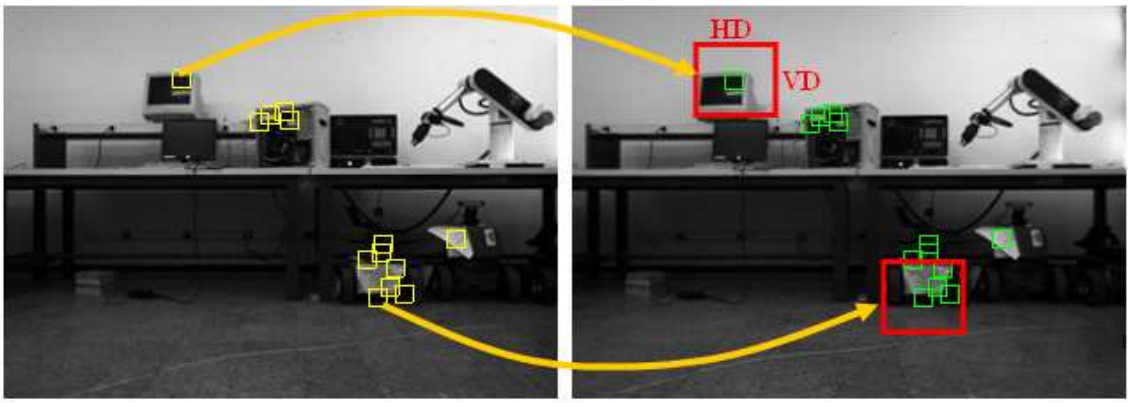
4.2.8.1 Stereovision Correspondence

In the last decade, a lot of computer vision work such as face recognition, stereo-ego motion, image mosaic…etc are proposed based on the extraction and matching of invariant features in an image, stereo image or successive images [27, 100]. Images acquired by stereo cameras have particular characteristics which can be used to speedup features matching step.

Looking for a correspondence in the entire image is computationally heavy and many useless distances are calculated. For example, if we assume M_i number of features in image I_i and M_j number of feature in image I_j then the complexity of the matching algorithm using Euclidian distance is $M_i \times M_j \times 128 \times 127$ summations and $M_i \times M_j \times 128$ multiplications. This huge number of operations which should run between two frames, makes the matching algorithm time-consuming and not very suitable for real time applications.

In this part of our work, we propose an approach to reduce the complexity of the feature matching process between stereo images using stereo vision constraints. Rather than looking for feature correspondence in the entire image, the area of search will be reduced to regions of interest (ROIs) limited by the horizontal and vertical disparities (HD and VD).

For example, to find the correspondent of a feature $p_i(x_i, y_i)$ in image I_i (image left), we will look into the region of centre (x_i, y_i) at image I_j (image right) limited by the vertical and horizontal maximum disparity VD and HD (respectively) as shown in Figure (4.22). Good feature matching obtained with significant reduction of the computation time is shown in Figure (4.23) with $DistRatio = 0.1$.



a) Image left b) Image right

Figure.4. 21 Features correspondence

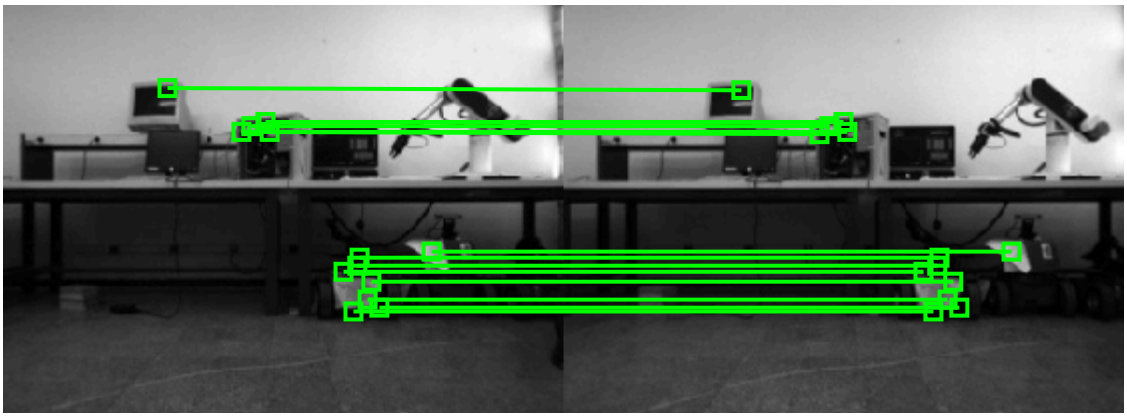


Figure.4. 22 Feature matching

4.2.8.2 Maximum Horizontal Disparity HD_{max}

Maximum Horizontal Disparity (HD_{max}) can be estimated as follows: Assume a point M of coordinates (x, y, z) in the world frame and its projection on the image left and right planes are (u_l, v_l) and (u_r, v_r), respectively. f is the focal distance and b is the baseline (distance between the two cameras). By triangle similarity as shown in Figure (4.24), we get:

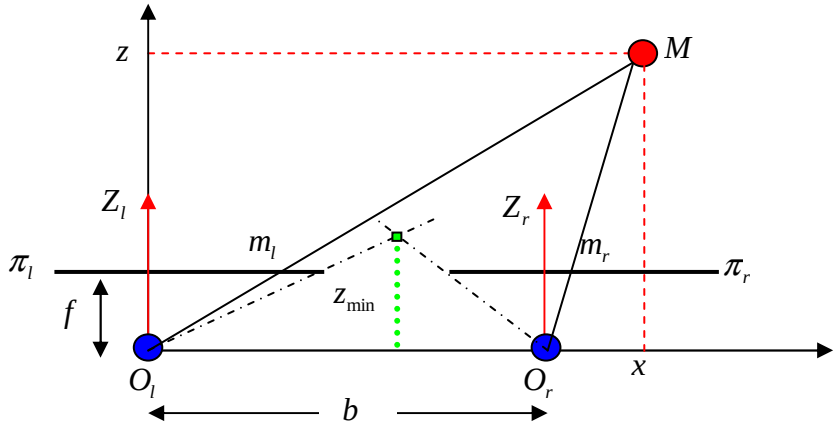


Figure.4. 23 Triangulation principle

$$\frac{f}{z} = \frac{u_l}{x} = \frac{u_r}{x-b} \tag{4.7}$$

$$\frac{f}{z} = \frac{u_l - u_r}{b} \tag{4.8}$$

However the disparity is defined by:

$$d = u_l - u_r \tag{4.9}$$

From Equation (4.8) and (4.9) we can obtain:

$$z = \frac{f \cdot b}{d} \tag{4.10}$$

From Equation (4.10), and for calibrated stereo-cameras, the depth z of the point M depends only on the disparity. From the same equation, we can conclude that the maximum disparity is linked with the minimum depth $HD_{max} = d_{max} = \frac{f \cdot b}{z_{min}}$ where z_{min} is the depth of the nearest point observed by both cameras (left and right).

4.2.8.3 Maximum Vertical Disparity $HV_{\max}$

The estimation of the vertical disparity is based on the epipolar geometry. Assume a feature p_l with coordinate (u_l, v_l) in the left image and its correspondent in the right image p_r with coordinate (u_r, v_r). Then, the equation of the epipolar line is given by:

$$p_r^T \cdot F \cdot p_l = 0 \quad \textbf{(4.11)}$$

where F is the fundamental matrix, given by:

$$F = (M_r^{-1})^T \cdot E \cdot M_l \quad \textbf{(4.12)}$$

M_l and M_r are the intrinsic parameter matrix for camera left and camera right respectively [100], E is the essential matrix.

If we put $F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$, then Equation (4.11) becomes:

$$u_l u_r f_{11} + u_l v_r f_{21} + u_l f_{31} + v_l u_r f_{12} + v_l v_r f_{22} + v_l f_{32} + u_r f_{13} + v_r f_{23} + f_{33} = 0 \quad \textbf{(4.13)}$$

and the epipolar equation will be given by:

$$v_r = A \cdot u_r + B \quad \textbf{(4.14)}$$

$$A = -\frac{u_l f_{11} + v_l f_{12} + f_{13}}{u_l f_{21} + v_l f_{22} + f_{23}} \quad \textbf{(4.15)}$$

$$B = -\frac{u_l f_{31} + v_l f_{32} + f_{33}}{u_l f_{21} + v_l f_{22} + f_{23}} \quad \textbf{(4.16)}$$

$$u_l - \frac{HD_{\max}}{2} \le u_r \le u_l + \frac{HD_{\max}}{2} \quad \textbf{(4.17)}$$

if $A > 0$ then:

$$A(u_l - \frac{HD_{\max}}{2}) + B \le Au_r + B \le A(u_l + \frac{HD_{\max}}{2}) + B \quad \textbf{(4.18)}$$

$$A(u_l - \frac{HD_{\max}}{2}) + B \le v_r \le A(u_l + \frac{HD_{\max}}{2}) + B \quad \textbf{(4.19)}$$

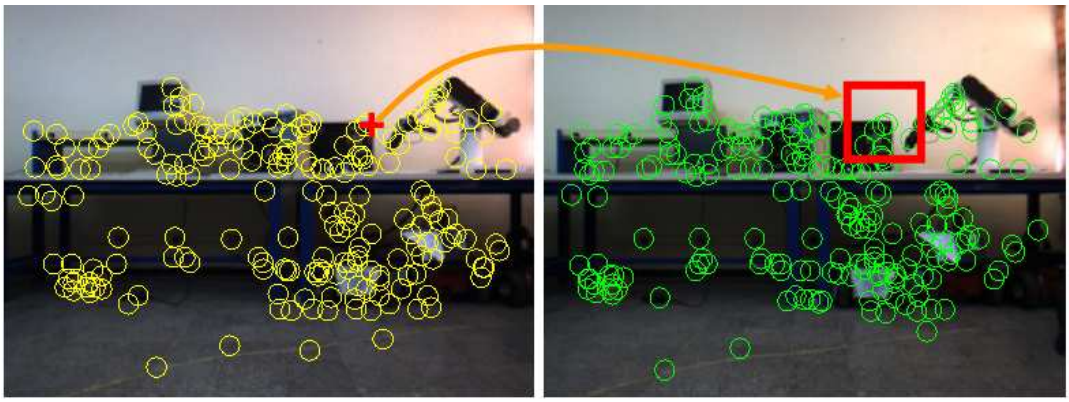
Then, we conclude that maximum vertical disparity is given by:

$$VD_{\max} = A(u_l + \frac{HD_{\max}}{2}) - A(u_l - \frac{HD_{\max}}{2}) \quad \textbf{(4.20)}$$

$$VD_{\max} = \begin{cases} A \cdot HD_{\max} & if \quad A > 0 \\ -A \cdot HD_{\max} & if \quad A < 0 \end{cases}$$

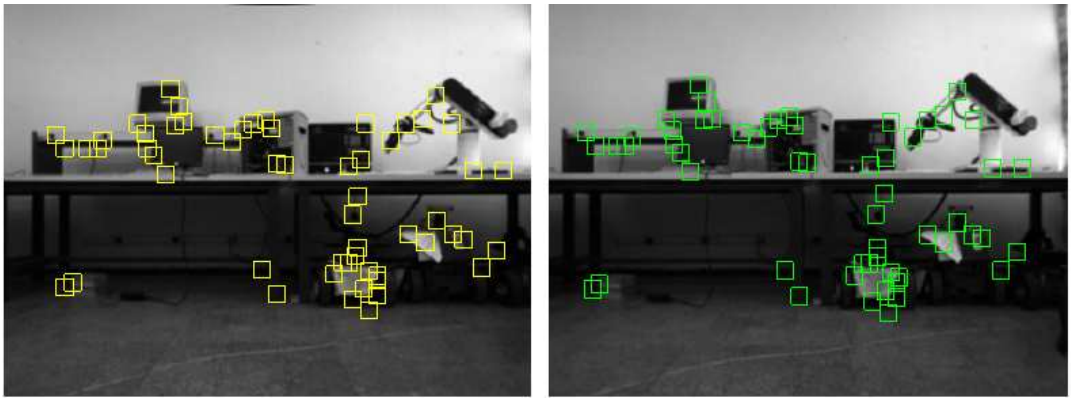
$$VD_{\max} = |A| \cdot HD_{\max} \tag{4.21}$$

An example using $DistRatio = 0.2$ is illustrated in Figure (4.25, 4.26 and 4.27). Figure (4.25) shows the extracted feature using adaptive SIFT with a disparity window. A landmark from image left got 06 candidates only rather than 245 (number of features extracted in left image). Figure (4.26) and (4.27) show the results of features matching.



a) Right image 241 landmarks b) Left image 245 landmarks

Figure.4. 24 Features extraction



-a- -b-

Figure.4. 25 Features correspondence

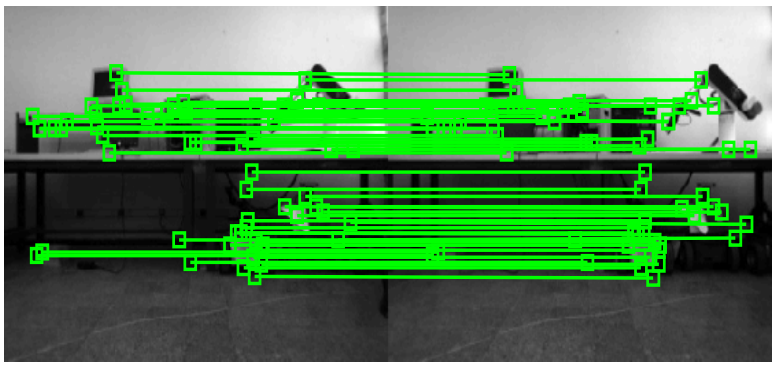


Figure.4. 26 64 good matching

If the left and right images are rectified then the matching problem becomes easier because in this case the vertical disparity $VD = 0$. Thus, the area of search of the feature will be reduced to one dimension space limited by the horizontal disparity HD.

4.3 Part II: Image Mosaicing

4.3.1 Introduction

The main objective of this section is to construct a large map for the UAV environment using a camera embedded in the aerial vehicle Figure (4.28). It is clear that in this kind of application, the acquired images present many changes (Geometric and photometric). Firstly the Aerial vehicle has 6 Degrees of freedom (6DOF), which leads to different geometrical transformations between current and next acquired image. Secondly the aerial vehicle navigates in natural environment; therefore the acquired images suffer from luminosity and contrast changes. To deal with these problems a mosaic algorithm based on robust features is presented in the following sections.

Figure.4. 27 Camera embedded on an UAV

4.3.2 Image Transformation

Images acquired by an Unmanned Aerial Vehicle can be affected by many transformations that we classify as geometric and photometric transformations:

• Geometric transformations

– Rotation

– Similarity (translation + uniform scale)

– Affine (scale dependent on direction)

valid for: orthographic camera, locally planar object

• Photometric transformations

– Luminosity

– Contrast

– Affine intensity change

4.3.3 Image Mosaicing

The first step in the image mosaicing algorithm is to extract and match ASIFT features between all the images. ASIFT features are located at scale-space maxima/minima of a difference of Gaussian function. At each feature location, a characteristic scale and orientation is established. This gives a similarity-invariant frame in which to make measurements. Although simply sampling intensity values in this frame would be similarity invariant, the invariant descriptor is actually computed by accumulating local gradients in orientation histograms. This allows edges to shift slightly without altering the descriptor vector, giving some robustness to affine changes. The vector of gradients is normalised, and since it consists of differences intensity values, it is invariant to affine changes in intensity. These performances of ASIFT features allow robust geometric and photometric image registrations, which are very important for image mosaicing algorithm.

4.3.3.1 Geometric Registration

In the context of this section, *geometric registration* refers to the process of obtaining a dense correspondence (or registration) between multiple views of a planar surface, or equivalently, between multiple views taken by a camera rotating about its optic centre. In both cases, the geometric transformation between any two such views is captured completely by an 8 degree-of-freedom planar projective transformation or homography.

Image homography will be used as a basic tool for estimating the motion that a UAV undergoes by using monocular image sequences. Summarising, a homography can be defined as an invertible planar transformation that applies lines into lines.

- Homography Matrix Properties

Some basic properties of the homography matrix are as follows:

– A homography between two planes is a linear transformation between three-dimensional homogeneous vectors y, represented by the 3×3 *H* matrix such as y = *H* x.

$$\begin{bmatrix} k\tilde{x} \\ k\tilde{y} \\ k \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.22}$$

– Given the homogeneous skill of the homography *H*, it can be multiplied by an arbitrary constant $k \neq 0$ and represent the same transformation. This means that the matrix *H* is constrained by eight independent parameters and a scale factor.

– The homography transformation between two images only holds exactly when the imaged scene is planar or almost-planar, a very common situation when the UAV flies at high altitude.

– The homography that relates two given images is computed from sets of matches between point features given by a feature tracker [84].

Depending on the frame-rate and the vehicle motion, the overlap between images in the sequence is sometimes small. This generates a non-uniform distribution of the features along the images. Hence, there may exist multiple solutions. However, using ASIFT algorithm as feature extractor improves the results, for two reasons: First, it detects suitable number of invariant and distinctive features. Second, extracted features face robustly with significant image changes (large image translation and rotation, scale change and photometric changes). These two advantages of ASIFT detector/descriptor permit to deal with the previous problems.

- Estimating Homography Matrix

Various methods for computing a planar homography between image pairs have been proposed, but they generally fall into two broad categories: Direct correlation methods, which compute the homography by maximizing photometric consistency over

the whole image. Feature based methods compute the homography from a sparsely distributed set of point-to-point correspondences.

The results presented in this chapter were generated using feature based registration methods. Feature based techniques have many significant advantages over their direct correlation counterparts in terms of computation speed, and the scope that they offer for the application of robust statistical methods for outlier rejection [85]. The planar homography has 8 degrees of freedom. Each point correspondence generates 2 linear equations for the elements of *H* and hence 4 correspondences are enough to estimate the homography directly. If more than 4 points are available, a least-squares solution can be found by linear methods. From the definition of *H*, we have:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \textbf{(4. 23)}$$

where "=" refers to the equality up to scale. Each inhomogeneous 2D point correspondence generates two linear equations in the elements of *H*.

$$\begin{aligned} x'(h_{31}x + h_{32}y + h_{33}) - h_{11}x - h_{12}y - h_{13} = 0 \\ y'(h_{31}x + h_{32}y + h_{33}) - h_{21}x - h_{22}y - h_{23} = 0 \end{aligned} \quad \textbf{(4. 24)}$$

Hence, *N* points generate *2N* linear equations, which may be arranged in a "design matrix".

$$AH=0 \quad \textbf{(4.25)}$$

The solution for *H* is the one-dimensional kernel of A, which may obtained from the Singular Value Decomposition (SVD).

For N>4 points, this equation will not have an exact solution. In this case, a solution which minimizes the *algebraic residuals*, *r = AH*, in a least-squares sense may be obtained, by taking the singular vector corresponding to the smallest singular value.

4.3.3.2 Mosaicing Algorithm

The proposed mosaicing algorithm is based on ASIFT detector/descriptor for a robust matching followed by estimating the homography for geometric registration. The following diagram describes the essential steps of the mosaicing based homography

approach. Figure (4.29) illustrates the stages to construct an image mosaic based on ASIFT detector/descriptor.

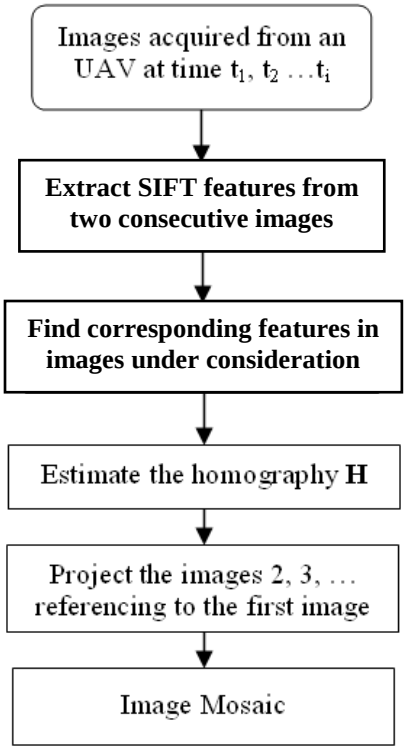
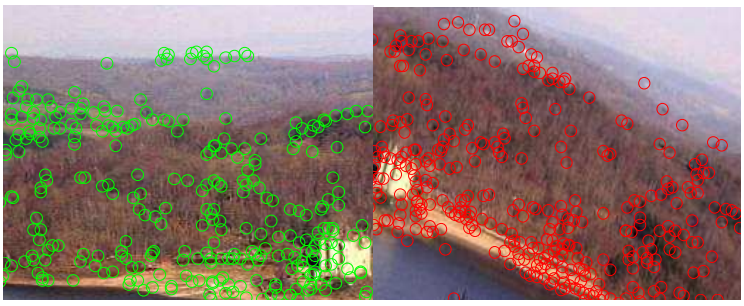
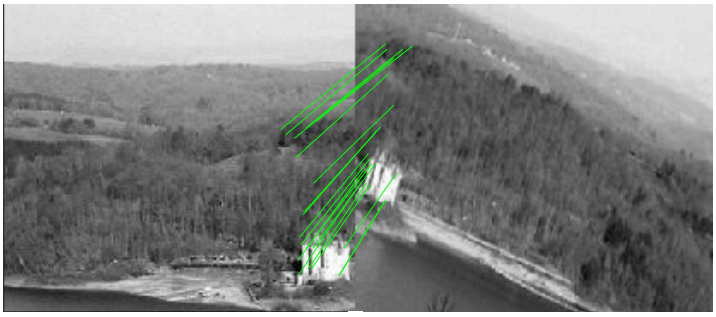


Diagram.2 Homography approach

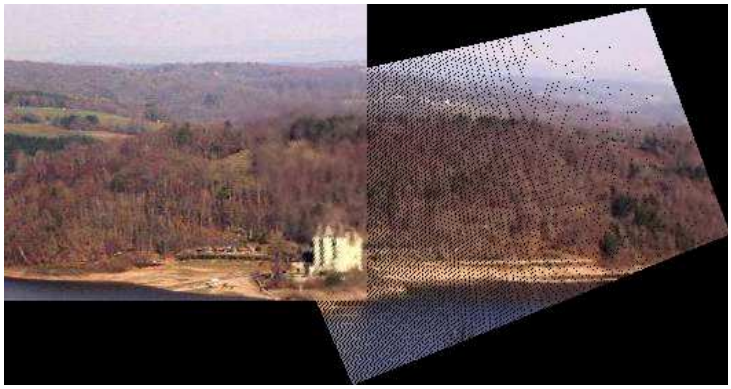
a b



c d



e



f

Figure.4. 28 a, b) two successive images seen by a camera embedded on an UAV, c, d) features detected in both images by SIFT detector, e) features matching using SIFT descriptor, f) Image mosaic (geometric registration).

The KLT (Kanade-Lucas-Tomasi) tracker also has largely been used to find correspondence between images for image mosaicing application. However, when the distance between the consecutive images increases, the KLT algorithm cannot track features successfully, Figure (4.30). Where, Figure (4.31) shows the result of matching obtained by the SIFT detector, 139 good matching are found, which is very suitable for a good mosaicing results Figure (4.32).

Image with considerable change in point of view

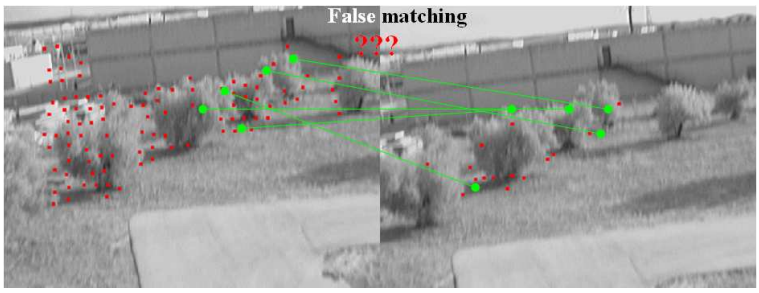


Figure.4. 29 Significant image change imply KLT false matching (failure)

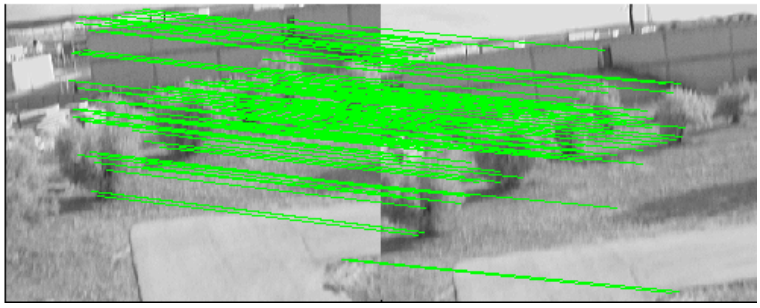


Figure.4. 30 ASIFT matching (139 good matching)

Figure.4. 31 Image mosaic based on ASIFT detector/descriptor

As can be seen from Figure (4.33), for small sampling time between consecutive images, the performance of KLT and SIFT are almost similar. However, when this sampling time increases the performance of KLT decreases (only few good matching) while SIFT maintains a suitable number of good matching, Figure (4.33).

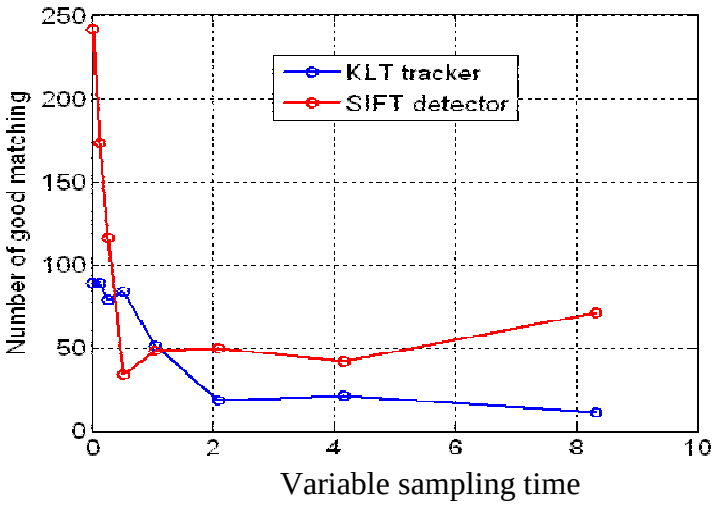


Figure.4. 32 Number of good matching detected by KLT & SIFT with different sampling time

Figure.4. 33 Image mosaic from a set of image acquired by a camera embedded on an UAV

From Figure (4.34) we can observe the good geometric registration of images even if big changes in point of view were made. Nevertheless, the photometric change can appear clearly in the doted square region. The following section will treat this problem to improve the quality of the mosaic image.

4.3.3.3 Photometric Registration

Photometric registration refers to the procedure by which global photometric transformation between images is estimated and compensated [85]. Examples of such transformations are:

- Global illumination changes across the scene.
- Intensity variations due to camera automatic gain control or automatic white balancing.

To solve these problems, first we should propose a suitable model of photometric transformation, and then we should identify the parameters of this model. The next section gives more details about the proposed approach.

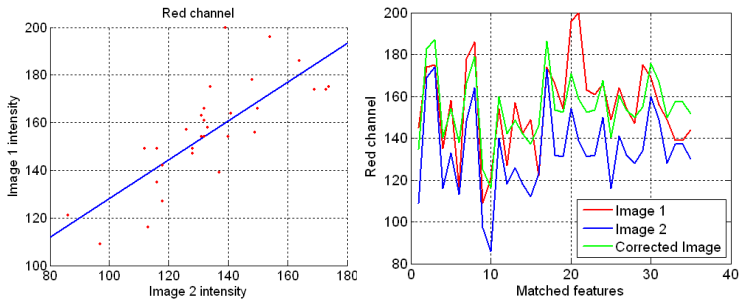
- The photometric model

The model treats each of the red, green and blue colour channels independently. Within each channel, the variation between the two images is modelled as a linear transformation, having 2 parameters: a multiplicative term α and an additive term β. Expressing the image pixels as 3-element vectors, the transformation can be written as:

$$\begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix} = \begin{bmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{bmatrix} \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} \beta_r \\ \beta_g \\ \beta_b \end{pmatrix} \tag{4.26}$$

and requiring 6 parameters in total. This simple model proves to be rich enough for the purposes described here. There is no apparent benefit in using a full affine transformation of the RGB colour-space.

The procedure for estimating the 6 photometric parameters requires the two images to first be accurately registered, using the method described in Section 4.3.3.1, and warped into alignment. The remaining differences between corresponding pixels in the aligned images can then hopefully be "absorbed" by the photometric model. Treating each channel separately, the estimation procedure for α and β is clearly a simple line-fit to the intensities of corresponding pixels (i_1, i_2), easily achieved by orthogonal regression Figure (4.35).



a b

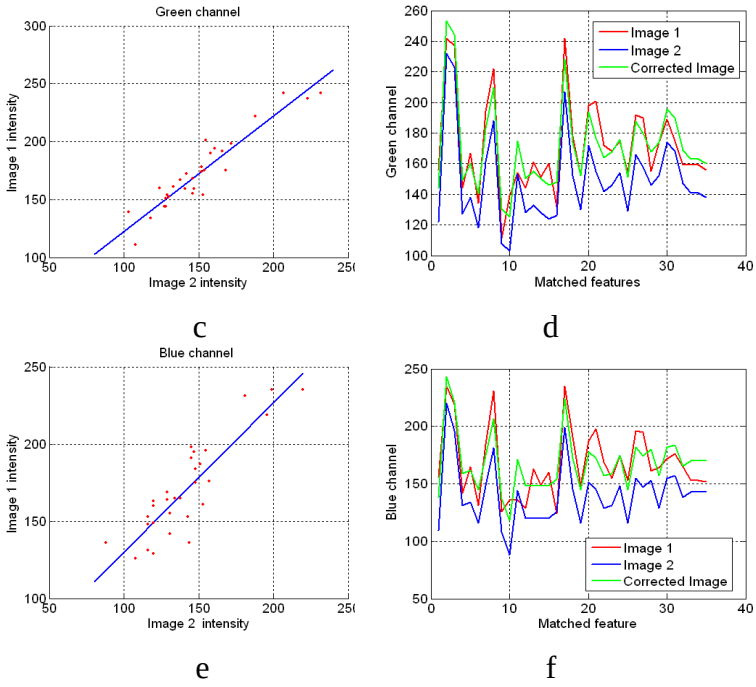


Figure.4. 34 Results of regression a), c) and e) estimation of α and β for the three channels red, green and blue, b), d) and e) Intensity of the three channels before and after correction.

Figure (4.36) shows the image mosaic of two images before photometric registration, the photometric change appears obviously, which affects the quality of the mosaic image. The photometric change is almost removed using the photometric registration as shown in Figure (4.37).

Figure.4. 35 Image before photometric registration (significant change of intensity)

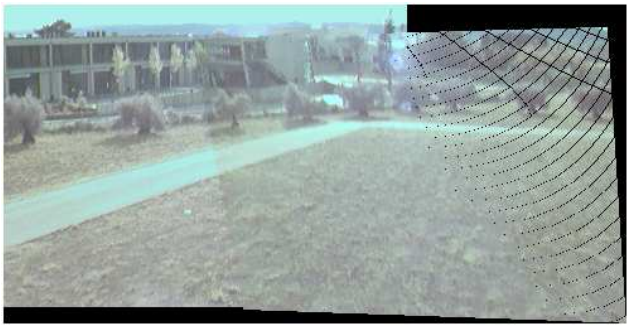


Figure.4. 36 Image after photometric registration (the intensity change is almost removed)

4.3.4 Super-resolution Algorithm

Super-resolution is an essential task to improve the quality of the mosaic image. Many algorithms are proposed in the literature [85, 87, 89]. In this work, we propose to use the iterative Papoulis-Gerchberg [86] algorithm using SIFT features. The super-resolution image should demonstrate an improvement in the perceived detail content compared to that of the low-resolution images. This will typically involve restoration of the high-frequency content, which in turn may require an increase in pixel density. Clearly, this definition is rather subjective. However, we would hope that in most cases, the improvement in perceived detail would be clearly visible to any observer.

4.3.4.1 Super-resolution Algorithm

Given a bunch of Low resolution (LR) images, Super-resolution involves two steps:

• Image Registration

• Projecting LR image values onto high-resolution grid

Most of the papers on Super-resolution try to solve these two problems to create a high resolution (HR) image from low resolution (LR) images. Although their approaches are different, the end goal is same. The following Figure (4.38) describes the two steps graphically. On the left hand side four LR images are shown. Motion estimation is used to estimate the pixel positions of the three images with respect to the first image. Note that these pixel positions would not be at integer values, but can take any real value. Once this information is calculated accurately, it is possible to project

this information on a desired high-resolution grid. Details of how these two problems are solved will be discussed in the following sections.

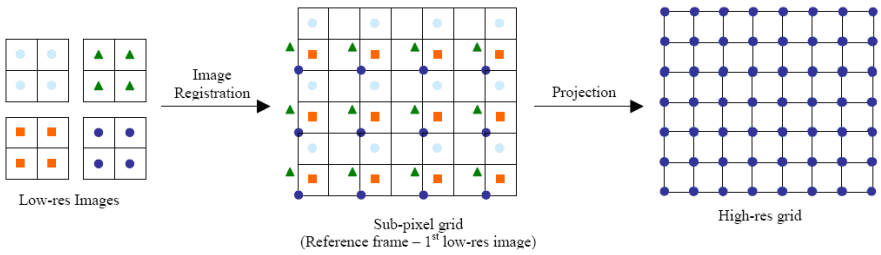


Figure.4. 37 Principal of the super-resolution algorithm

4.3.4.2 Papoulis-Gerchberg Algorithm

Papoulis-Gerchberg algorithm [87- 89] illustrated in Figure (4.39) can be described as follows:

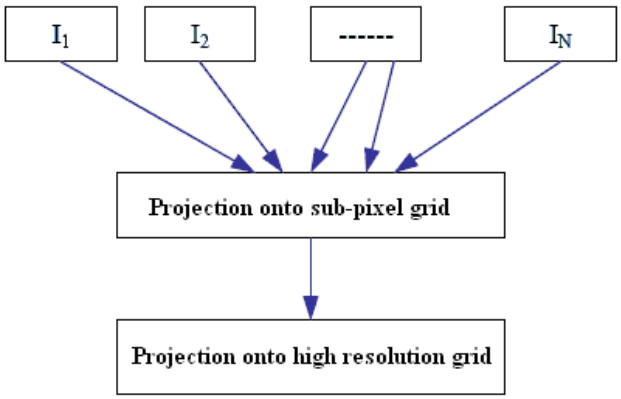


Figure.4. 38 Essential steps of Papoulis-Gerchberg Algorithm

where all the calculation of images ***Ii*** are performed to one reference image $\boldsymbol{I}_1$. This algorithm works by projecting HR grid data on the two sets (sub-pixel and HR grids) described above. The algorithm steps are:

• Form a HR grid. Set the known pixel values from the LR images (after converting their pixel position to the reference frame of first low-resolution image). The position on the HR grid is calculated by rounding the magnified pixel positions to nearest integer locations.

• Set the high frequency components to zero in the frequency domain.

• Force the known pixel values in spatial domain.

• Iterate.

Figure (4.40) shows an example of the Papoulis-Gerchberg algorithm using ASIFT features applied on four low resolution images to construct one high resolution image. As can be seen the quality of the high-res image is much better than the low-res images.

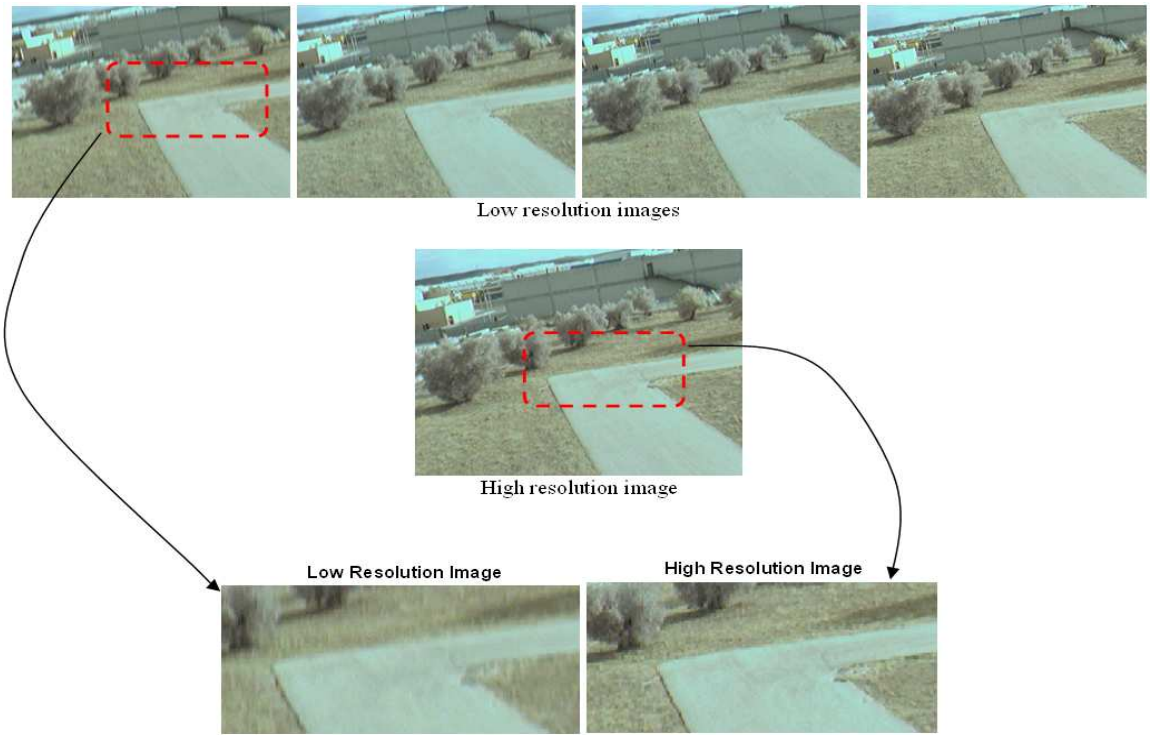


Figure.4. 39 Low and High resolution images using Papoulis-Gerchberg Algorithm based

4.3.5 Experimental Results (Mosaicing improved by Super-resolution)

Figure (4.41.a) presents a set of images acquired by a camera embedded on an Unmanned Aerial Vehicle (UAV). Figure (4.41.b) shows an image mosaic constructed from the acquired images using simultaneously the Papoulis-Gerchberg algorithm for super-resolution and the proposed mosaicing algorithm. The result is one large mosaic image with high quality Figure (4.41.b).

Figure.4. 40.a Images acquired by a camera embedded on an UAV

Figure.4. 41.b Image mosaic

4.4 Part III: New Feature Detector and Descriptor

4.4.1 Introduction

As mentioned in the first part of this chapter, Harris Corner detector, Harris Laplace detector and Scale invariant feature transform (SIFT) are some popular feature extractors proposed in the literature. A good feature extractor should provide a high rate of repeatability and suitable accuracy while requiring reasonable computational time. Repeatability of a feature extractor should be evaluated against geometric (Rotation, Scale change, Affine…) and photometric (Luminosity, Contrast, Blur…) changes in the image.

In this section, we propose to develop a new feature extractor suitable for unmanned vehicle visual navigation applications. This passes by looking at the following three main steps:

1- Highlight the proposed feature extractors in the literature and focus on the limits of each one.
2- Build a new feature extractor robust against geometric and photometric image transformation and overcome the limits of previous feature extractors.
3- Comparison of the new extractor with those proposed in literature using dataset containing most common image transformations.

4.4.2 Scale Wavelet Invariant Feature Detection

By studying the existing detectors and from published comparisons [80, 96], we can conclude that 1) Hessian-based detectors are more stable and repeatable than their Harris-based counterparts. Using the determinant of the Hessian matrix rather than its trace (the Laplacian) seems advantageous, as it fires less on elongated, ill-localised structures. 2) DoG approximation can accelerate feature detection but it decrease the accuracy of the algorithm. 3) The interpolation of extrema in image scale/space and the non maximum suppression is time consuming.

The proposed approach follows another alternative since we believe that a good significant feature can be detected if and only if the image is well analysed and presented. Therefore, by inspiring from the Human Visual System the best way to

represent an image in scale and space is the multi resolution analysis. This latter can be obtained by the 2D Discrete Wavelet Transform (DWT) [150]. Many authors tried to use DWT for corner detection. However, they use the DWT as a derivative tool in different direction without taking the advantage of the multi resolution aspects [83]. Therefore, we propose a novel detector scheme named ***SWIF*** (Scale Wavelet Invariant Feature). The proposed detector contains two main steps: First, the multi wavelet transformation of the image is calculated. Then, the extrema detection is achieved in scale, space and directions.

4.4.3 SWIF Detector Diagram

Based on the theory and implementation of 2D discrete wavelet transform [150], the wavelet pyramid construction and local extrema detection are presented.

4.4.3.1 Characteristics of 2D Discrete Wavelet Transform

Human visual characteristics are widely used in image processing. One example is the use of Laplacian pyramid for image coding. SIFT falls into the category that uses Laplacian pyramid for scale-invariant feature extraction [151]. On the other hand wavelet transform is a multi-resolution transform that repeatedly decompose the input signal into low-pass and high-pass components like sub-band coding [151, 152]. Wavelet-based scale-invariant feature extraction method does not increase the number of samples in the original image, which is the case of the Gaussian pyramid-based SIFT method. Wavelet transform can easily reflect human visual system by multi-resolution analysis using orthonormal bases [152]. Because the wavelet-based method does not increase the number of samples, computational redundancy is greatly reduced, and its implementation is suitable for parallel processing.

The discrete wavelets transform (DWT) decomposes an input signal into Low (L) and High (H) frequency component using a filter bank followed by decimation by a factor of 2 as illustrated in Figure (4.42). Daubechies wavelet, which characterises the filter bank, has important properties of orthonormality, linearity, and completeness. DWT will be repeated multiple times to multiple-level resolution of different octaves.

In order to construct the wavelet pyramid, we decide the number of Daubechies coefficients and approximation levels. The purpose is to extract salient points from any

part of the image in order to detect any change in the image at any resolution. A high wavelet coefficient (in absolute value) at a coarse resolution corresponds to a region with high global variations. The properly chosen length of the Daubechies wavelet and the number of the approximation levels provides the optimum local key points or features.

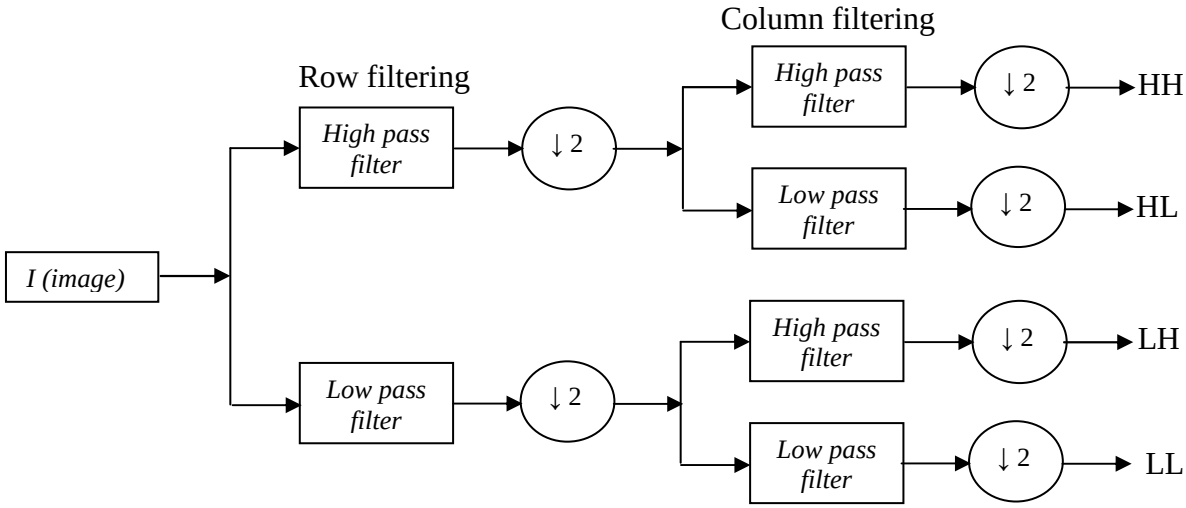


Figure 4. 42 DWT decomposition

4.4.3.2 Scale Space Feature Extraction

The two main steps of Scale Space Feature Detector are:

i) Multi Wavelet Image Transformation

The original image I will be represented using multiple Daubechies (db) wavelet [150] as follows:

$$[A_k \quad H_k \quad V_k \quad D_k] = DWT(I, \text{"}db_k\text{"}) \tag{4.27}$$

k is the Daubechies wavelet order. A_k, H_k, V_k and D_k are approximation image, horizontal, vertical and diagonal details respectively.

Equation (4.27) will be repeated with different values of k at different levels to construct the multi resolution pyramid of the image.

ii) Extrema Detection in Scale, Space and Directions

In this step local extrema (minimum and maximum) will be detected in scale and space following the three directions (horizontal, vertical and diagonal). Local maxima

and minima of the multi-resolution wavelet pyramid are detected as follows: each sample point is compared to its eight neighbours in the current image and its nine neighbours in the scale above and below. It is selected only if it is larger than all its neighbours or smaller than all of them. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few checks. This step is similar to that used by Lowe in SIFT algorithm [81].

Figure (4.43) shows the architecture of the proposed algorithm. The wavelet pyramid contains many octaves (wavelet levels). At each octave: 1) the original image is decomposed using multiple Daubechies (*db*) wavelet. 2) Extremas of wavelet details are determined following the three directions (horizontal, vertical and diagonal).

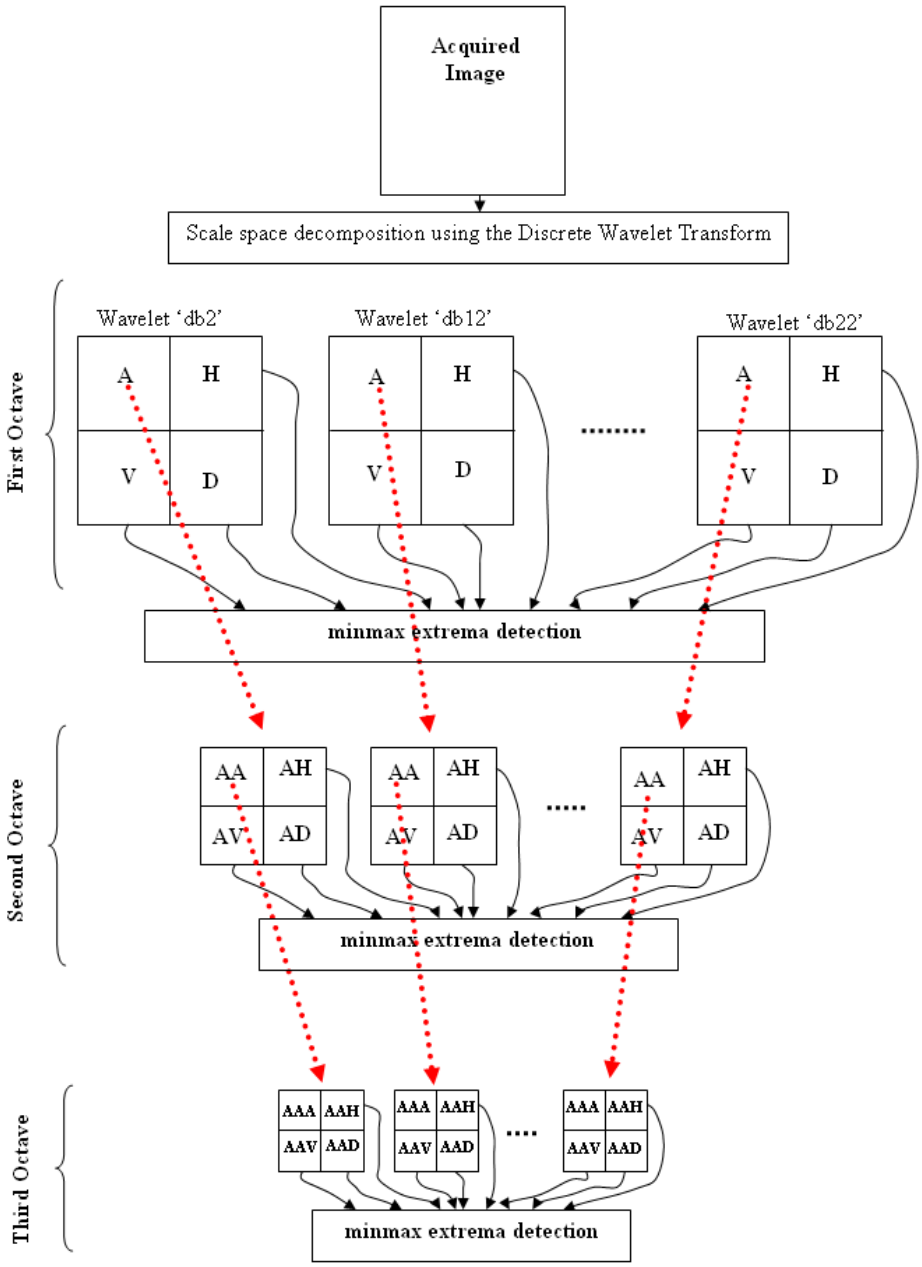


Figure.4. 43 Features extraction using SWIF detector

Figure (4.44) and (4.45) respectively show the results of the DWT decomposition and the difference of wavelet pyramid obtained by two Daubechies wavelets 'db_2' and 'db_{10}';

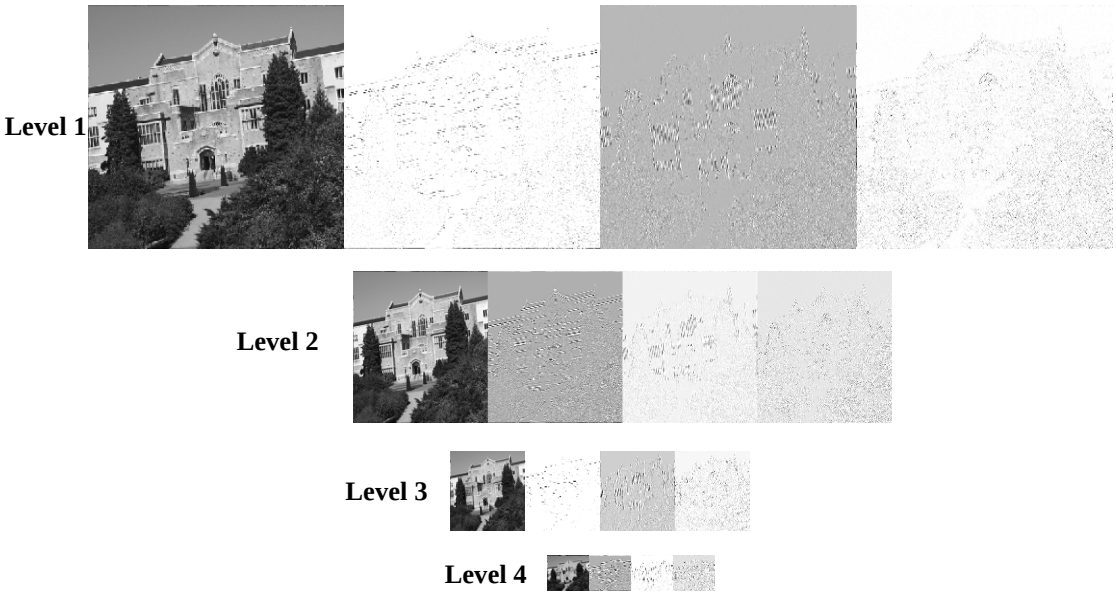


Figure.4. 44 Four levels wavelet decomposition from left to right, approximation image, horizontal, vertical and diagonal details,

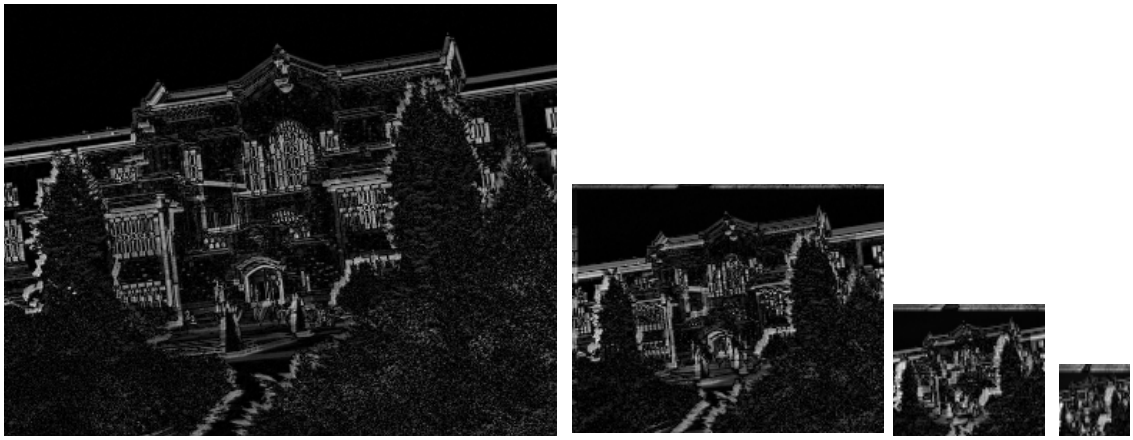


Figure.4. 45 Difference of wavelet images

4.4.4 Advantages of SWIF Detector

Our proposed approach (SWIF) has the followings advantages:

1- Features extracted from wavelet coefficients are invariant to image rotation because a feature is considered if and only if it is detected following the three directions. This means that a detected feature by SWIF is in the same time

horizontal, vertical and diagonal detail. As consequence it will be redetected at any image rotation.

2- Wavelet coefficients (detail coefficients) are the results of filtering using high pass filter. Thus they are not affected by luminosity change.
3- Using the multi-resolution analysis to construct the pyramids of wavelet make our detector robust to scale changes.
4- Detected features are only those which present significant horizontal, vertical and diagonal details in multiple wavelet levels. This means that there is no chance for features situated on edges to be selected. Also, we do not need another step for nonmaxima suppression as done in SIFT [81].
5- The use of orthogonal wavelets reduces the information redundancy in the same level and also between levels.
6- The 2D DWT ensures a parallel calculation of horizontal, vertical and diagonal details which is promising for real time implementation.

4.4.5 Comparison with other feature detectors:

An important characteristic of a feature extractor is its repeatability. The repeatability is defined as the capability of a detector to find the same features (up to a threshold) in two different images of the same scene. The following figures show a comparison between SIFT, Harris, Harris—DWT and the proposed SWIF. Harris—DWT is a corner detector using DWT coefficient rather than second moment matrix [75]. The repeatability of detectors is validated using image dataset available at [153]. It contains different kind of image transformations (Geometric and Photometric). Dataset of viewpoint change, photometric change, image blur, rotation and scale change are respectively presented in Figures (4.46, 4.48, 4.50 and 4.52). Repeatability against these changes are shown in Figures (4.47, 4.49, 4.51 and 4.53) respectively.

Figure.4. 46 Dataset viewpoint change

From Figure (4.47), SIFT and Harris show high rate of repeatability (~80%) with small change of viewpoint. However, this repeatability decreases when the viewpoint angle increases. The SWIF detector gives acceptable repeatability rate (~60%) with small viewpoint angle. More importantly this repeatability is almost maintained when the viewpoint angle increases. As can be seen with viewpoint angle of 60 deg the SWIF repeatability is much better than SIFT and Harris (SWIF is evaluated using different thresholds Th1<Th2< Th3). This propriety is very important for feature matching.

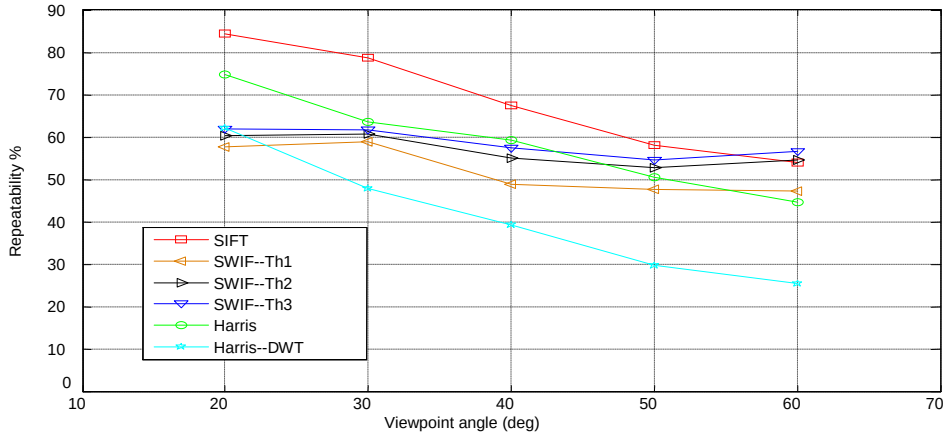


Figure.4. 47 Repeatability of detectors to viewpoint change

Figure.4. 48 Dataset Photometric change

From Figure (4.49), SWIF shows high rate of repeatability (~90%) with a small photometric change than SIFT and Harris. However, the repeatability of all detectors decreases when this photometric change increase. We should mention that this photometric change is not the sample luminosity or contrast change (addition or multiplication by a parameter) but it is a kind of nonlinear light changes introduced by varying the camera aperture.

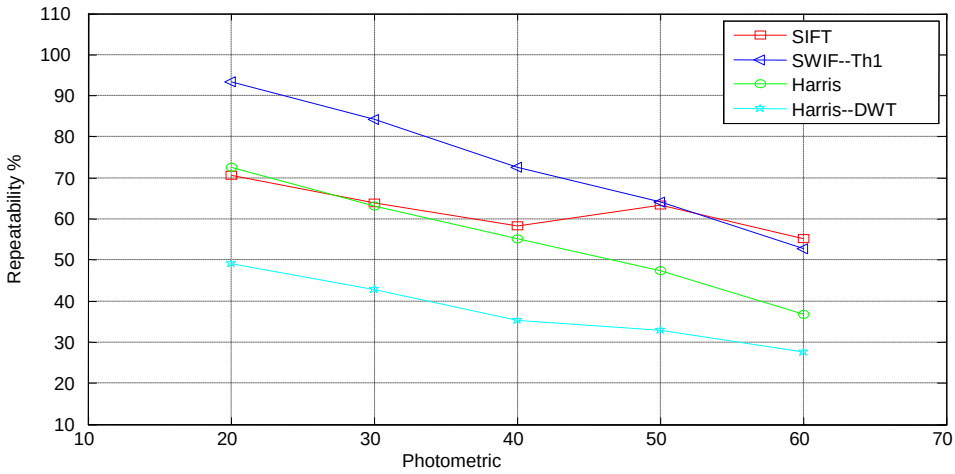


Figure.4. 49 Repeatability of detectors to photometric change

Figure.4. 50 Dataset image blur

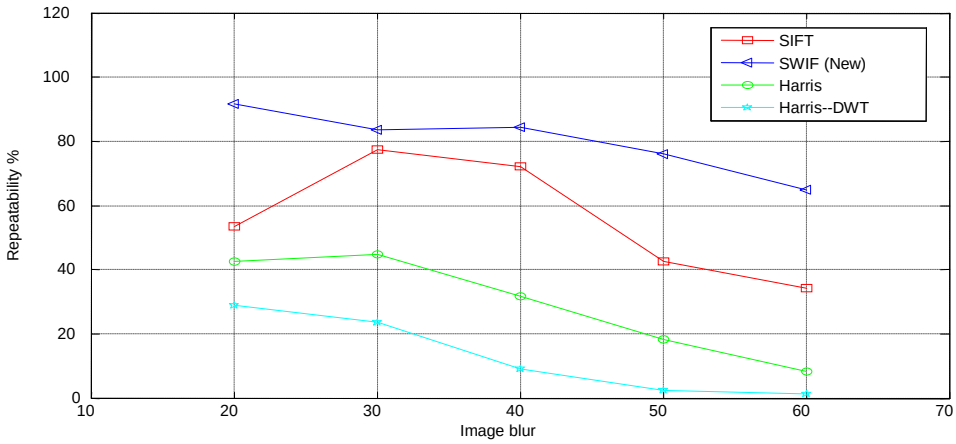


Figure.4. 51 Repeatability of detectors to image blur

Figure (4.51) shows that SWIF presents high rate of repeatability (~90%) in comparison to SIFT and Harris. This high rate is maintained even with significant camera focus

changes. This high repeatability rate of SWIF can be explained by the robustness of the wavelet multi-resolution analysis.

Figure.4. 52 Dataset rotation and scale change

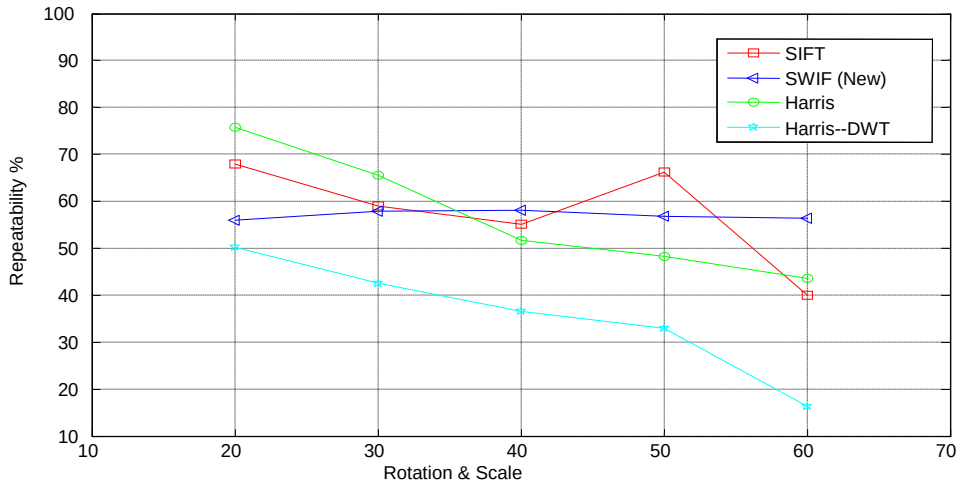


Figure.4. 53 Repeatability of detectors to rotation and scale change

From Figure (4.53) SIFT and Harris show high rate of repeatability (~70%) with small image rotations and scale changes. However, this repeatability rate decreases with significant image rotation and scale change to a repeatability of (~40%). Conversely, the SWIF detector gives acceptable repeatability rate just under (60%) with small image rotation and scale change. This repeatability is roughly maintained when image rotations and scale changes increase. We can also observe that the proposed SWIF performs much better than SIFT and Harris when the image is affected by significant rotation and large scale change. As we said before this is very important to avoid the detection of unstable features which is time consuming.

Figure.4. 54 Dataset luminosity change

Figure (4.55) shows the repeatability rates of the different detectors facing 50% of luminosity change presented in Figure (4.54). As can be seen SIFT and SWIF show high repeatability rate comparing to Harris detector which is partially robust to luminosity change.

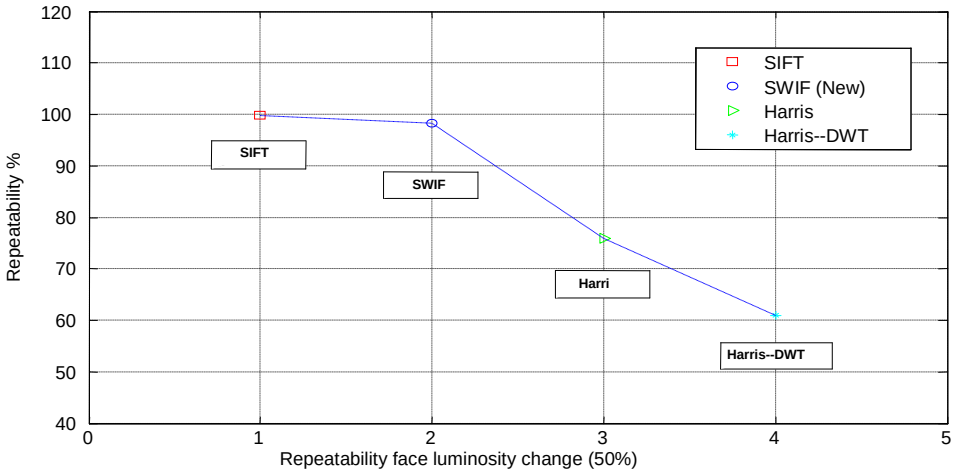


Figure.4. 55 Repeatability of detectors to luminosity change

4.4.6 SWIF Features Stability

This section presents performance details of the SWIF detector in terms of features stability at each pyramid wavelet level using different thresholds.

As can be seen from Figure (4.56, 4.57 and 4.58), the most stable features are detected in the (high) third level of the wavelet pyramid. Figure (4.59) shows how the threshold value can affect the number of extracted features. A suitable value of the threshold *Th* between (15 and 25) is determined by experiment. Figure (4.60) shows the results obtained by applying the SWIF detector on a sequence of indoor images. More

than 80% of features are re-detected which is very important for our future tasks (VSLAM, image mosaic,…).

Finally, Figure (4.61) shows the robustness of the SWIF detector against significant scale changes. Yellow points are detected at level 2 of the wavelet pyramid while green points are detected at level 3. As observed from this figure, our objective is achieved as enough number of features is re-detected.

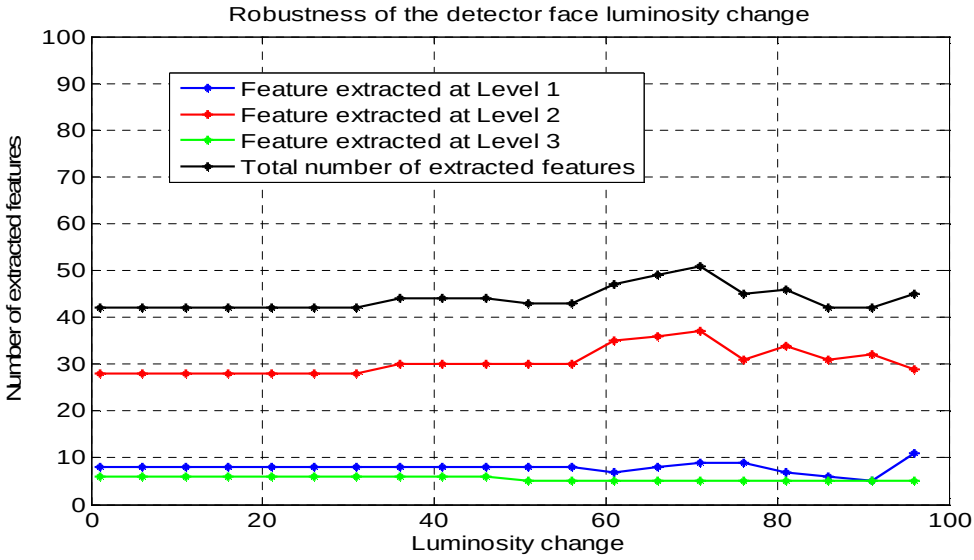


Figure.4. 56 Repeatability of detectors (at each wavelet level) to luminosity change

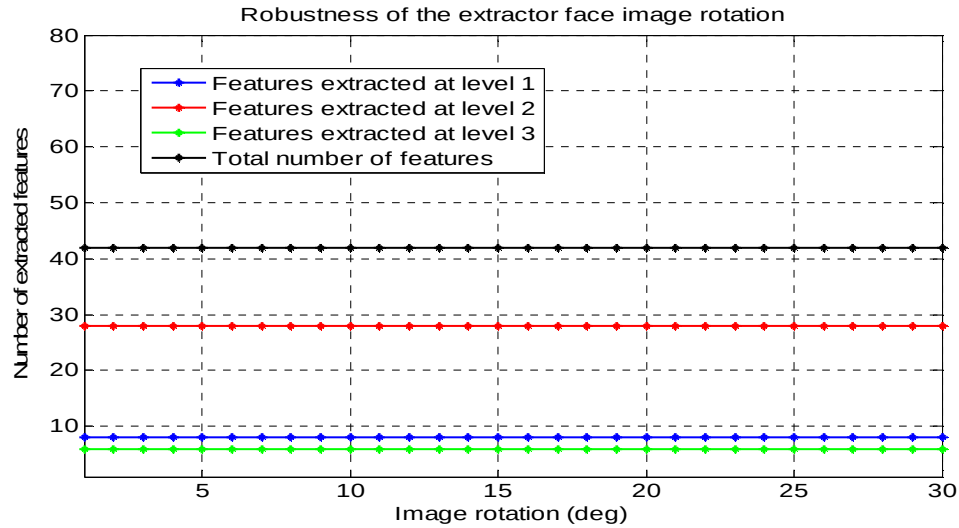


Figure.4. 57 Repeatability of detectors (at each wavelet level) to image rotation

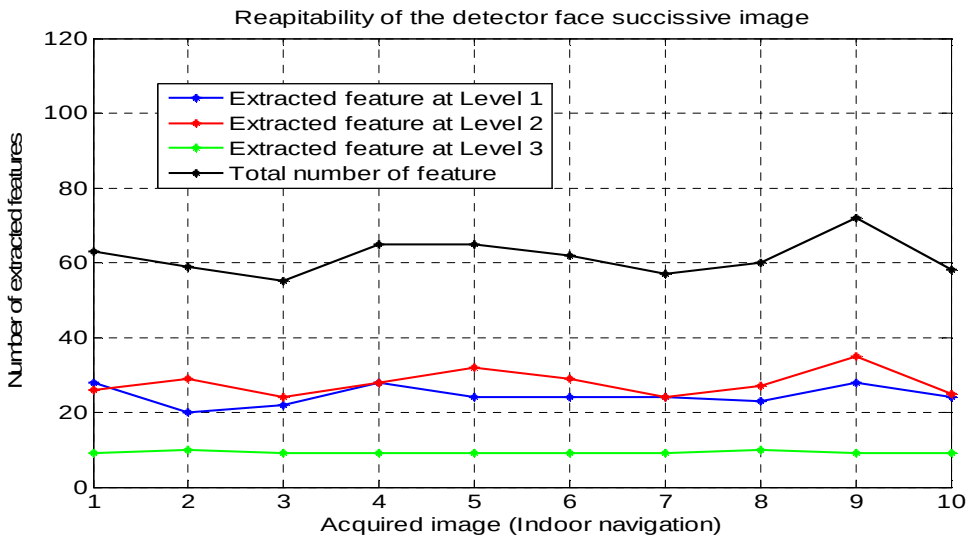
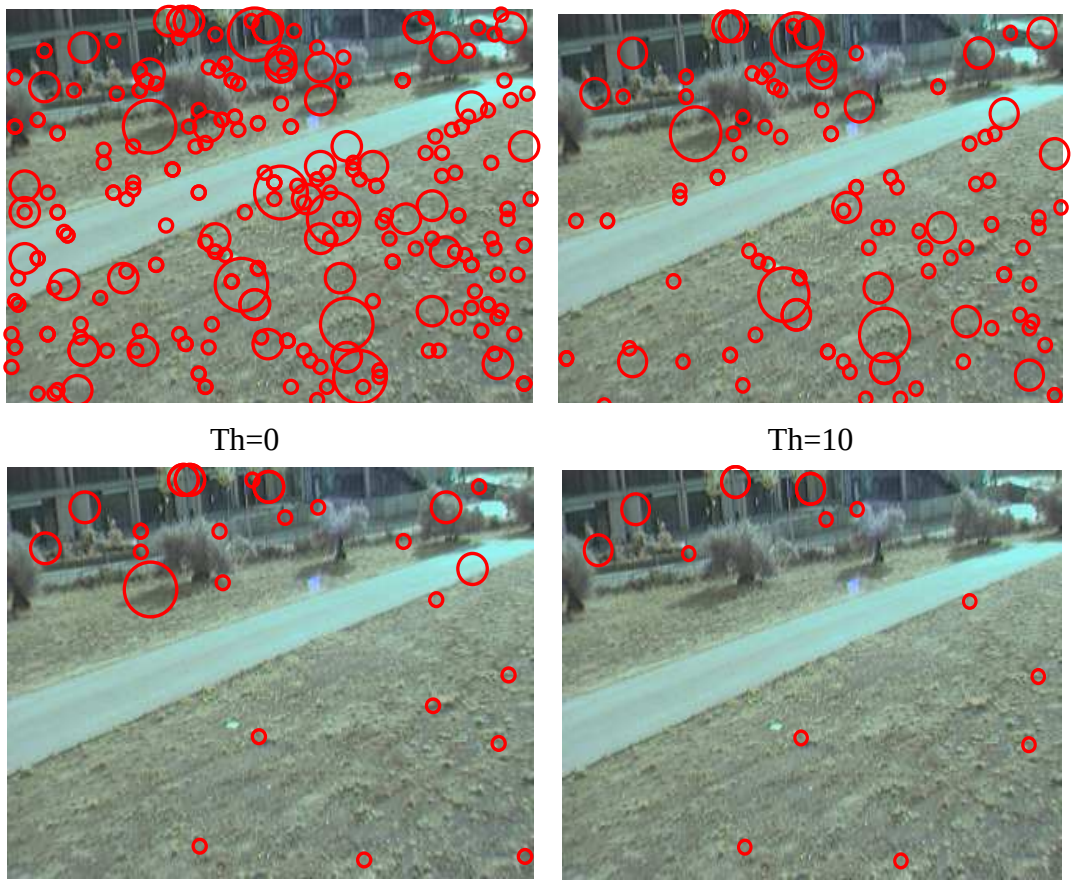


Figure.4. 58 Repeatability of detectors (at each wavelet level) using outdoor images



Th=0 Th=10

Th=15 Th=20

Figure.4. 59 Number of the extracted feature when we change the threshold "Th"

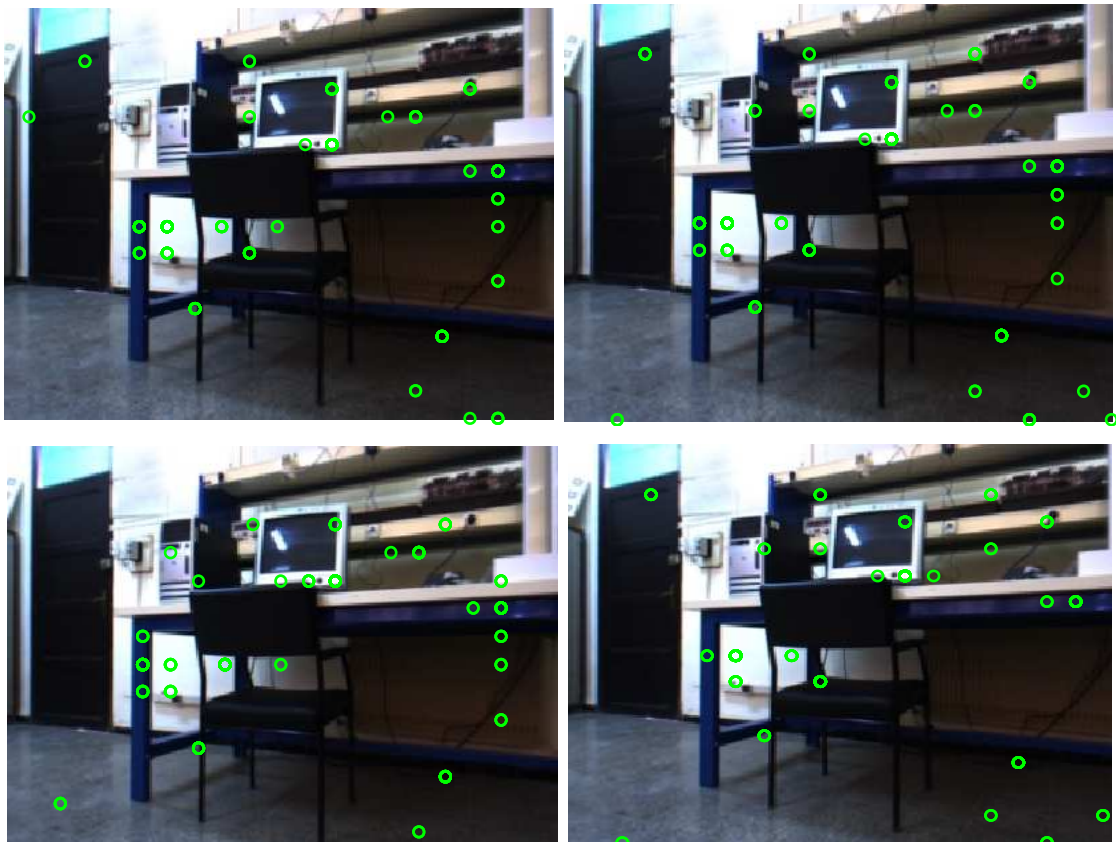


Figure.4. 60 Application of the new feature extractor on successive image More than 80% of features are re-detected

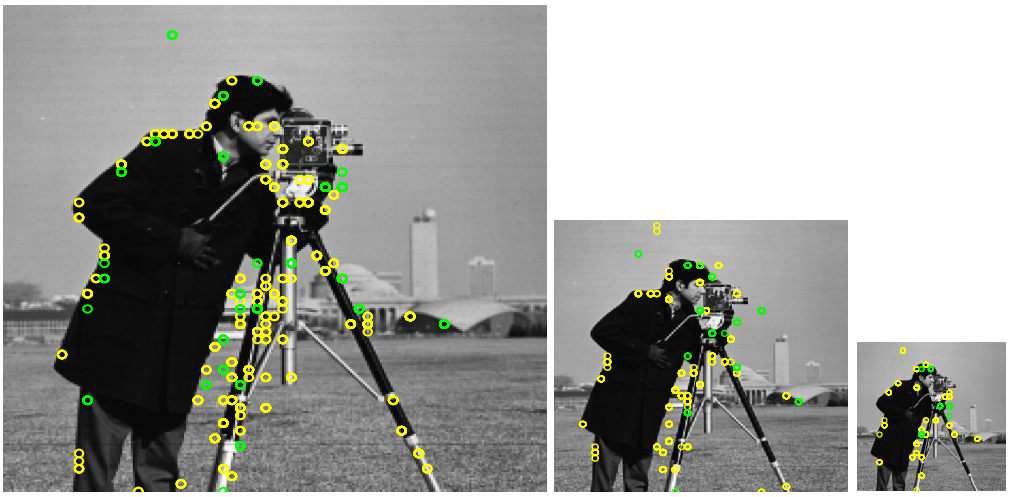


Figure.4. 61 Robustness face scale change, enough number of same features is detected

4.4.7 SWIF Descriptor

Features extracted by the new Scale Wavelet Invariant Feature (SWIF) detector have the following form: $f_i = [u_i, v_i, k_i]$, where (u_i, v_i) are the image coordinates of the feature f_i and k_i is the wavelet level where the feature f_i is detected.

Based on the result given by the SWIF detector a fast and invariant features descriptor is designed.

The SWIF descriptor is designed to be invariant facing geometric transformations (Rotation, Scale change, affine…) and photometric changes (Luminosity, contrast, image blur…). The size of the SWIF should be proposed to avoid significant increase of the computation time of the matching step. Lowe [81] proposes a descriptor of size 128 based on the difference of gauss image gradient. Lowe descriptor is not only so long but also its conception requires a significant computation time. In the following section we present two main steps of orientation assignment and descriptor construction for SWIF algorithm:

4.4.7.1 Orientation Assignment

For each feature $f_i = [u_i, v_i, k_i]$ we should find a local orientation based on the wavelet transform of the original image I as follows:

a- A square region R_{fi} of size $S/2^k$ is selected around the specific feature. S is the size of the region, k is the wavelet level. When $k = 0$, a suitable value of $S = min(size(I))/2^4$ is used. For example, if the size of the image I is 256×256 then the suitable value of S is 16x16.

b- Use the horizontal and vertical wavelet coefficient to calculate the orientation of each pixel of the selected region R_{fi}, as follows:

Assume that H_k and V_k are the horizontal and vertical details of the image I at a level k. Then, the orientations $\theta_{i,j}^f$ and the norm $N_{i,j}^f$ of each pixel from the region R_{fi} are given by:

$$\begin{cases} \theta_{i,j}^{f} = \arctan\left(\dfrac{V_k(i,j)}{H_k(i,j)}\right) \\ N_{i,j}^{f} = \sqrt{(V_k(i,j))^2 + (H_k(i,j))^2} \end{cases} \qquad i,j \in 1...S/2^k$$

c- Use histogram orientation to calculate the dominant orientation as done in [81]. If we use an orientation step $\Delta\theta = 18^\circ$ ($\Delta\theta = 9^\circ$), pixel orientations are distributed on 21 (41) bins. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the feature. Peaks in the orientation histogram correspond to dominant directions of local regions. The dominant orientation Θ_f is the maximum of these peaks, and N_f is the norm of that orientation [81].

d- Orientation update: after that the dominant orientation Θ_f is determined then all the orientations will be recalculated in reference to Θ_f as follows: $\theta_{i,j}^{f} = \theta_{i,j}^{f} - \Theta_f$. This step makes our descriptor invariant to image rotation.

4.4.7.2 Descriptor Construction and Evaluation

SWIF descriptor vector is proposed to contain the norms $N_{i,j}^{f}$ reoriented following the dominant orientation Θ_f and weighted with a Gaussian window centred on (u_i, v_i), as done in [81]. This Gaussian window allows the descriptor to give more importance to the pixels near to the feature location. The descriptor vector is then normalized in order to provide robustness against image contrast change.

Our descriptor is built on simple concepts since we are limited by the real time constraints. Because the attended application is Visual SLAM, hence, extracting and matching only few but robust features is enough and very suitable. To evaluate our descriptor, we propose to use two performance criterions first, the matching score which is the rate between the number of good matching and the number of extracted feature. Second, the variation of the number of good correspondences. Below we explain the advantage of each criterion.

i) Matching score (*MS*)

$$Matching\ \ Score(MS) = \frac{Number\ of\ \ Good\ Matching(NGM)}{Number\ of\ \ Extracted\ \ Features(NEF)}$$

$$NEF = NGM + Number\ of\ \ False\ Matching\ (FM) + Number\ of\ \ \ Non\ Matched\ Features(NNMF)$$

The maximum value of the matching score (MS) is 1 (100%), which means all the extracted features are well matched. This is a rarely case. Following the equation above, when the number of good matching increases the matching score increases too. We can then conclude that the matching score does not depend on the number of good matching only but it takes in consideration the number of extracted features.

ii) Variation of Number of Correspondences (VNC)

$VNC = Number\ of\ \ Correspondences\ (T + \Delta T) - Number\ of\ Correspondences\ (T)$, which means the variation of number of extracted features between two image views of the same scene. This variation will give us an idea about the stability of good matching which is very important for many computer vision applications, as the VSLAM we are dealing with in this thesis. The most stable feature descriptor corresponds to a steady state VNC.

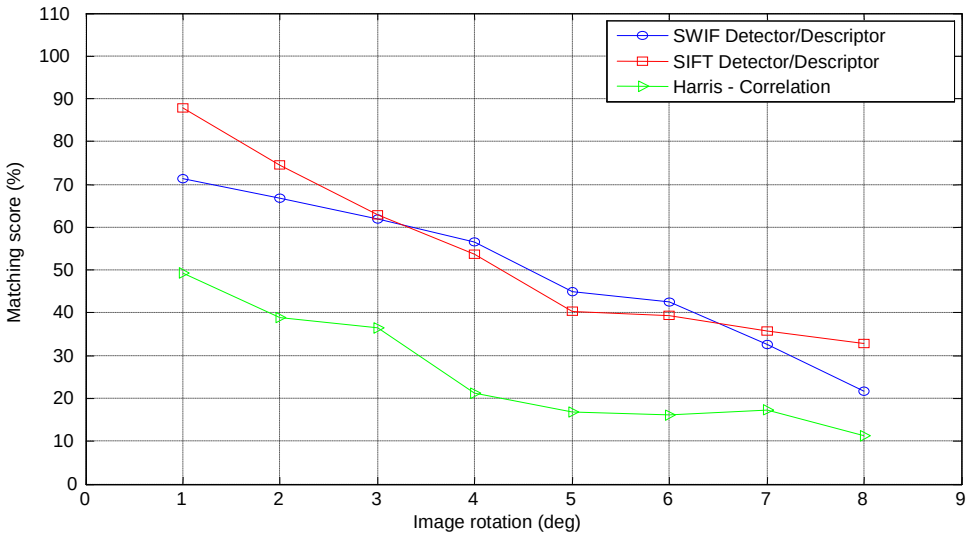
4.4.7.3 Results and Discussion

The new SWIF descriptor is validated using the dataset used previously in Section 4.4.5. The used images are affected by many geometric transformations (Image rotation, scale change and view point change) and photometric transformations (Luminosity and contrast change). Two techniques of feature extraction and matching are compared with SWIF. First, Harris based Correlation approach, where features are extracted using Harris Corner Detector then the matching is done using spatial correlation, Second, the Scale Invariant Feature Transform (SIFT) detector and descriptor.

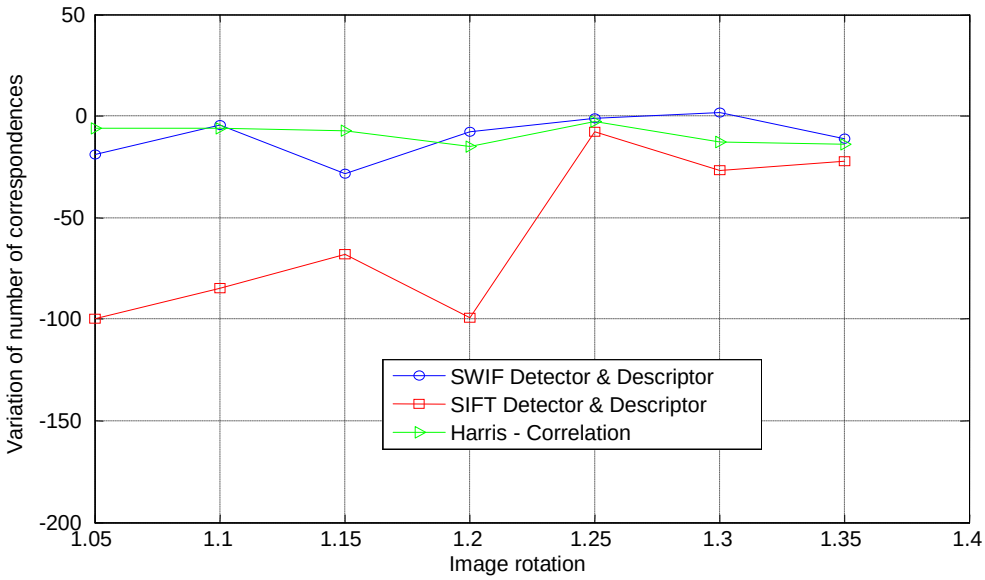
i) Image rotation

Figures below show the performance of matching when facing image rotation. As can be seen from Figure (4.62.a), quite similar performances of SIFT and SWIF are obtained, while Harris based correlation approach perform less well. Big number of extracted features by SIFT is not suitable for real time application but also the important

decrease of the number of correspondences is unwelcome for robust VSLAM application. From Figure (4.62.b), we observe that the *Variation of Number of Correspondences* (*VNC*) is steady for the SWIF comparing to the SIFT which confirm the good performances of the proposed technique.



a- Matching score

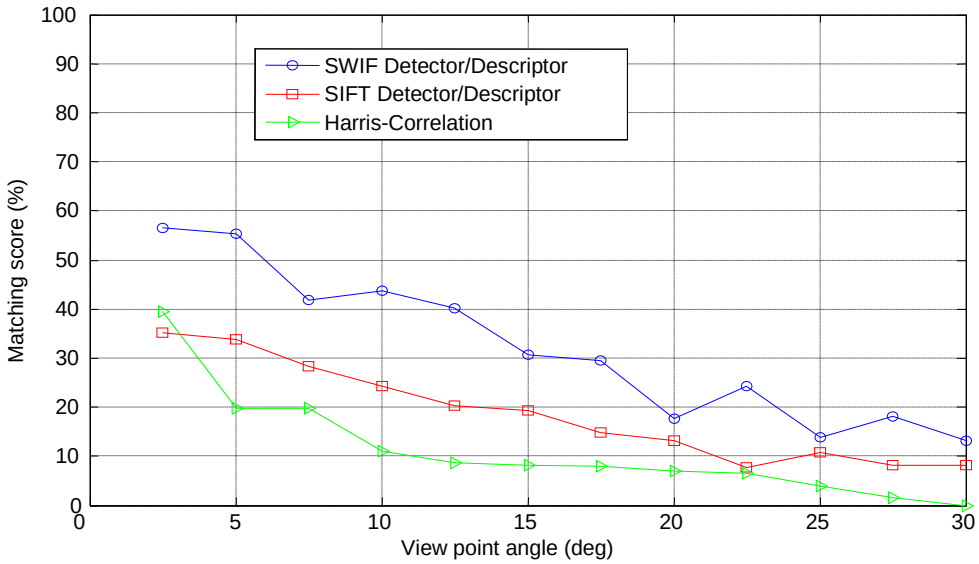


b- Variation of Number of Correspondences (VNC)

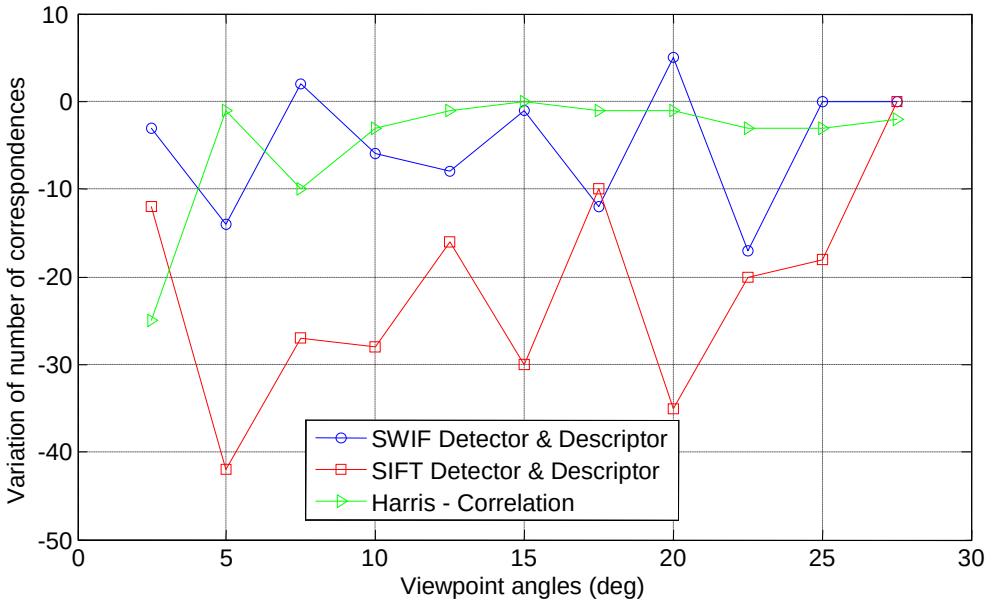
Figure.4. 62 SWIF descriptor performances face image rotation

ii) Image viewpoint change

Figures below show the performances of matching when facing image viewpoint angle change. As can be seen from Figure (4.63.a) SWIF score matching is much better than that obtained by the SIFT descriptor and Harris based correlation matching. Even if the number of good matching given by SIFT is bigger than the number obtained by the SWIF, but we should mention that the latter is more efficient than the former for the simple reason that in SWIF more than 50% of features are well matched. In the other hand in SIFT about 30% of features only are well matched. From this short analysis we can conclude that the proposed SWIF descriptor shows more efficiency in less computation time. Figure (4.63.b) confirms the results, as can be seen the *VNC* in SWIF is steadier than that obtained by SIFT.



a- Matching score

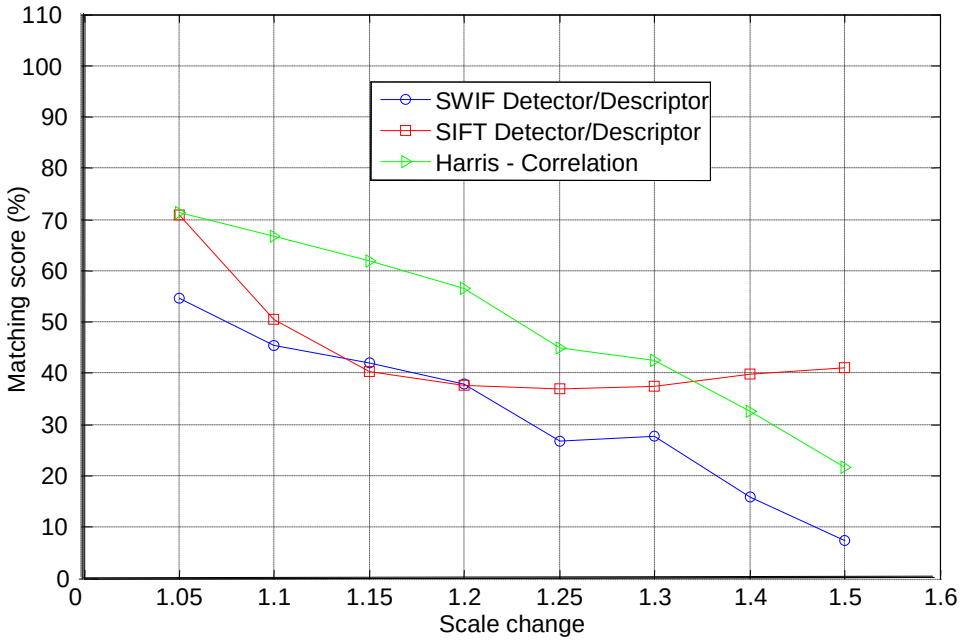


b- Variation of Number of Correspondences (VNC)

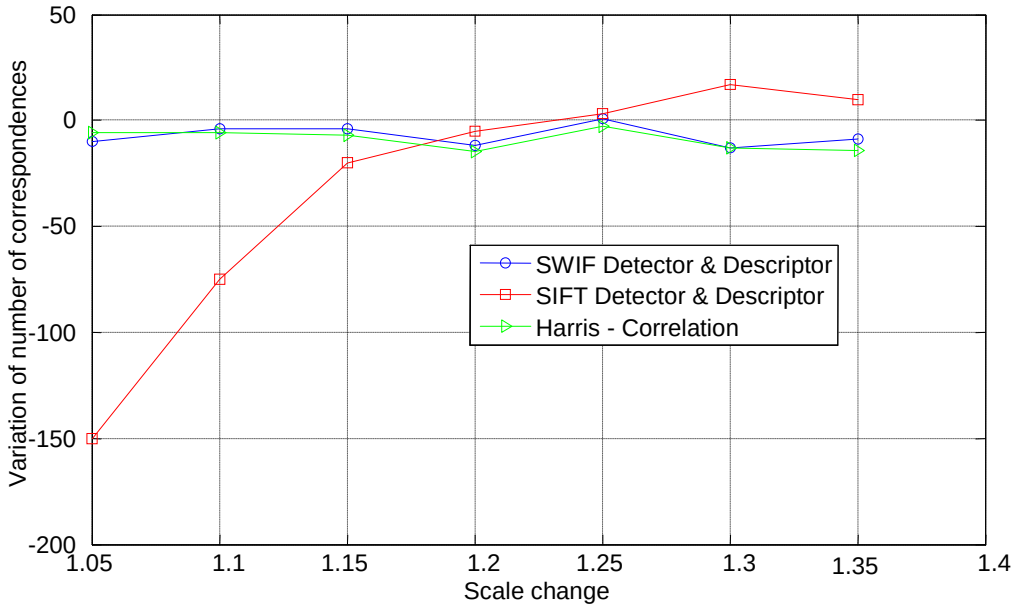
Figure.4. 63 SWIF descriptor performances face image viewpoint angle change

iii) Image scale change

From Figure (4.64.a), a quite similar performances are obtained with SWIF, SIFT and Harris based correlation. However with significant scale change, SWIF and Harris based correlation matching score decreases when SIFT maintain good performance. This can be explained by the large scale exploration done by the SIFT algorithm (five scale octave, each octave contains a large scale range), while SWIF uses three octaves with three wavelet levels in each octave. It is clear that SIFT provides significant robustness to scale changes at the cost of a heavy computation time. From Figure (4.64.b), we can observe that the *VNC* is steady for the SWIF comparing to SIFT which confirm the previous results.



a- Matching score



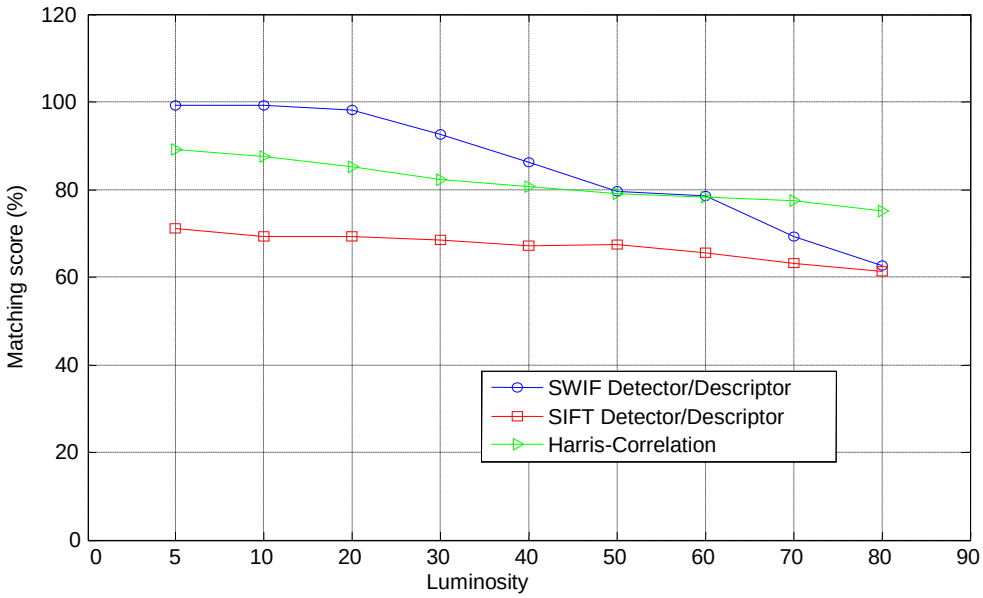
b- Variation of number of correspondences

Figure.4. 64 SWIF descriptor performances face image scale change

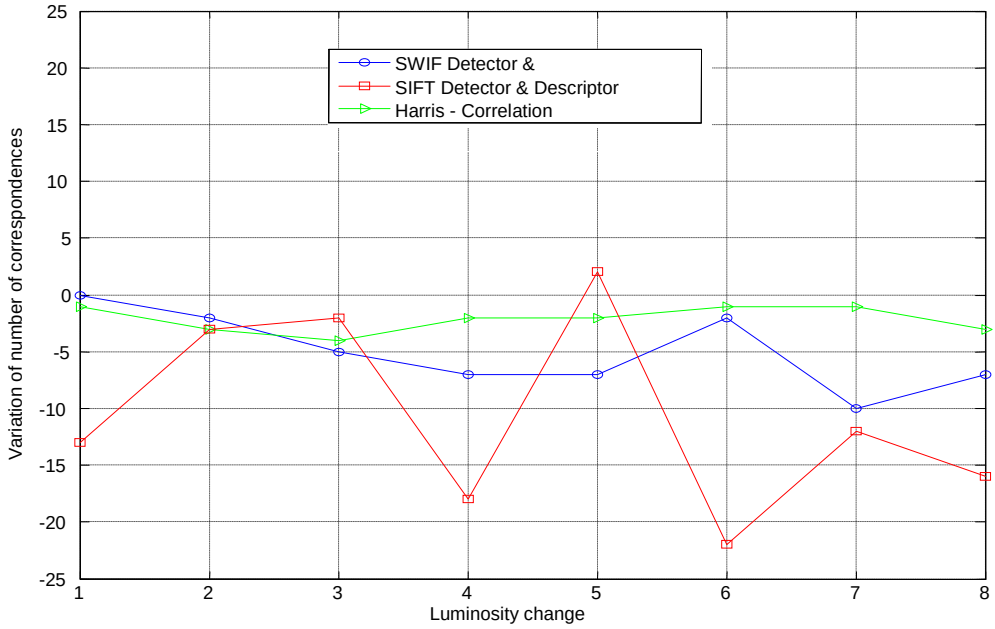
iv) Image luminosity change

Figures below show the performances of matching when facing image luminosity changes. As can be seen from Figure (4.65.a), quite similar good performances of SWIF

and correlation approach are obtained. From Figure (4.65.b), the *VNC* is steady for the SWIF comparing to the SIFT. This result shows the stability of the number of good matching using SWIF which is very important for VSLAM application.



a- Matching score

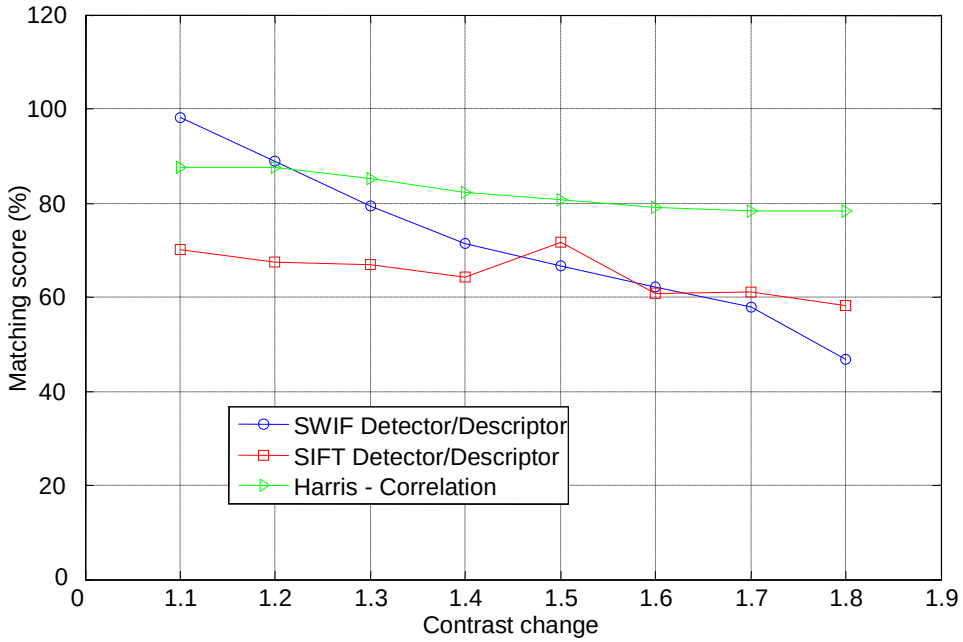


b- Variation of Number of Correspondences (VNC)

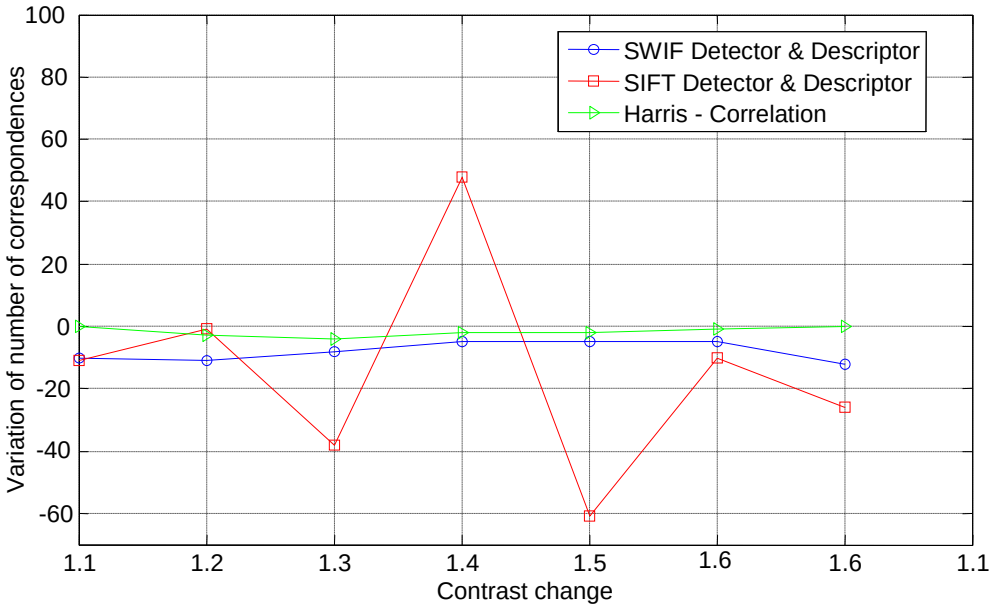
Figure.4. 65 SWIF descriptor performances face image luminosity change

v) Image contrast change

The figures below show the performances of matching against image contrast changes. Figure (4.66.a) shows quite similar good performances of SWIF, SIFT and Harris based correlation approach. Figure (4.66.b) shows that the Variation of Number of Correspondences (VNC) is steady for the SWIF and Harris based correlation comparing to the SIFT. Feature detected and matched by SWIF are almost stable which is very important for VSLAM application, in the other hand, significant variation of the number of SIFT good matching is observed in Figure (4.66.b). The unstable features not only they are time consuming but they are not useful for features based VSLAM.



a- Matching score



b- Variation of Number of Correspondences (VNC)

Figure.4. 66 SWIF descriptor performances face image contrast change

4.4.8 Computation Time Comparison

The three codes SWIF, SIFT and Harris based correlation are run under MATLAB (version 7.1.0.246, R14, Service pack 3) in a computer core duo processor CPU 1.85 with 2GB RAM and applied on gray scale images of size (256x256).

The computation time for SWIF detector and descriptor depends mainly on two parameters: the size of the description window ($ms \times ms$) and the orientation sampling ($bin \in [0,\ 360]$). This latter is used for orientation assignment which mean that the number of selected orientation is $1+\frac{360}{bin}$. Below is an evaluation of the SWIF computation time using different values of ms and bin.

For $ms = 8$ pixel

bin (deg)	3	9	18	30	36	90
Time	5.76 s	5.29 s	5.50 s	6.46	6.58 s	1.37 s
False matching rate	30 %	00 %	00 %	00%	00%	16.6 %

Table 4. 1 computation time vs *bin*

From this table, we can see that the number of bins does not have a big effect on the computation time. However, it affects highly the performance of the descriptor. A small value of *bin* makes the algorithm very sensitive to noise. As a result, significant false matchings will be obtained. On the other hand, a big value of *bin* makes a bad description of the feature and many features will have similar descriptors which lead to too many false matchings. Thus suitable values of *bin* should not be very small and should not be too big. As can be seen 9, 18 until 36 degrees are suitable values of bin. These latter will be tested using different sizes of description window ($ms \times ms$).

For $bin = 9^{\circ}$

ms	4	6	8	12	16
Time	5.47 s	5.02 s	5.21 s	5.44 s	6.61 s
False matching rate	12.5 %	00%	00%	00%	00 %

-a-

For $bin = 18^{\circ}$

ms	4	6	8	12	16
Time	5.37 s	5.11 s	5.59 s	5.98 s	7.65 s
False matching rate	16.6 %	5.9 %	00 %	00 %	00 %

-b-

For $bin = 36^{\circ}$

ms	4	6	8	12	16
Time	6.29 s	6.30 s	6.64 s	6.73 s	8.25 s

-c-

Table 4. 2 computation time vs ***ms***

Table (4.2.a, b and c) show the computation time and the performances of the SWIF algorithm using suitable values of *bin* with different values of *ms*. From these tables, we can see that the suitable values of *bin* and *ms* that give the best performances in short time are: $bin = 9$ and $ms = 8$ leading to a computation time of $Time = 5.21 s$ without false matching.

The comparison between SWIF and SIFT is made when a quite similar number of features are detected.

	Detector & Descriptor	Observation
SWIF	5.21 s	Descriptor size = 41
SIFT	7.49 s	Descriptor size = 128

Table 4. 3 Computation time SWIF vs SIFT

The suitable parameters of SWIF algorithm are then $bin = 9$, $ms = 8$ and descriptor size = 41. As can be seen the new feature detector/descriptor performs much better in time than the classical SIFT proposed by Lowe, Table (4.3). This decrease of computation time is the results of two main contributions: first, using the wavelet pyramid which ensures a parallel computation in less time than the DoG pyramid. Second, the use of short descriptor based on wavelet parameters.

4.5 Conclusion

Build a reliable map is an essential task of UAVs. In this chapter, we provided robust tools to construct a high resolution map of the environment. In the first part of this chapter we looked at providing solutions for the main problem of map building by proposing feature detection and matching. First we contributed into improving the classic SIFT by introducing an adaptive scale factor parameter, which ensures a steady performance for different kind of images (low and high frequency images). By proposing $\infty - norm$ distance criterion, a robust feature matching operation is achieved. We also developed, using some experimental analysis results, a criteria based on the scale rate to avoid the Adaptive SIFT features instability, which affects highly the matching performance.

In the second part of this chapter we presented an image mosaicing algorithm based on geometric registration, followed by photometric registration to deal with contrast and luminosity changes. We also proposed Papoulis-Gerchberg super-resolution algorithm using SIFT features leading with the mosaicing algorithm to construct a large mosaic image with high quality. Experiments were successfully carried out with real image data obtained by a camera setup onboard an unmanned helicopter.

In the third part of this chapter a new Scale Wavelet Invariant Feature (SWIF) detector/descriptor is proposed. SWIF seems to be a promising solution for VSLAM problem. Its robustness using a short descriptor makes the difference comparing to the proposed feature extractors available in the literature.

Chapter 5

Robust 3D Visual SLAM

CHAPTER 5

Robust 3D Visual SLAM

5.1 Overview

In this chapter the 3D Visual SLAM is presented, beginning by a literature review of the VSLAM problems, the Airborne VSLAM then is developed with loop closing detection. After that the robust *NH*∞VSLAM algorithm is proposed with observability and consistency analysis. A new map management approach is also proposed in this chapter in order reduce the algorithm complexity. Finally, simulation and experimental results with comparison between EKF and *NH*∞ VSLAM are presented.

5.2 Introduction

One of the current SLAM challenges is related Airborne SLAM and more specifically Airborne 3D VSLAM. This challenge requires a scalable representation, robust data association algorithms, consistent estimation techniques and fusion of different sensor modalities.

Traditionally in robotics, most research uses use data from odometer, laser-range-finder and sonar sensor. Currently, cameras are becoming competitive alternatives due to their low cost and the rich information content they provide.

Solving SLAM with monocular / stereo vision systems is still an open research problem to address efficiently and naturally many autonomous systems in real life applications. Davison, in [93], proposed a vision-based SLAM approach, which uses active stereo head and odometer sensing to estimate the location of a robot in planar regions. In [94], Davison has looked at the localisation and mapping problem using data from a single passive camera and using an EKF filter. However, SLAM architectures based on Extended Kalman Filter are very sensitive to outliers and increase computational complexity in a square form according to the number of features. To solve these problems, Nir and Bruckstein, [95], proposed a particle filter based SLAM approach to estimate camera (vehicle) poses. To be used for dynamic real time

applications. Particle filters based SLAM approaches are still under improvement, [98-99], in order to achieve efficient algorithms suitable for real time application.

Another important issue, which has been neglected in a number of SLAM implementations, is the convergence analysis of the SLAM algorithm. Only very few analytical results on the convergence and essential properties of SLAM algorithms are available in the literature. Dissanayake in [90], provided some convergence properties of the Kalman Filter SLAM. Lower bounds on the absolute accuracy of the map and the vehicle location were also obtained by an EKF based SLAM algorithm [90]. Kim in [101], provided some further analyses on the asymptotic behaviour of one dimensional Kalman Filter SLAM problem. Having said that, all the proofs presented in [100-101], deal only with simple linear formulations of the SLAM problem in contrast to most SLAM implementation requirements in terms of process and observation model nonlinearities. In the past few years, a number of researchers have demonstrated, [102-103], that the lower bound for the map accuracy presented in [90] is violated and the EKF SLAM produces inconsistent estimates due to errors introduced during the linearisation process. While some explanations of these inconsistency phenomena have been reported, mainly through Monte Carlo simulations a thorough theoretical analysis of the nonlinear SLAM problem became available only very recently [104].

To date and to the best of our knowledge, very few robust and efficient solution has been proposed to solve the Airborne 3D VSLAM problem taking into account the high nonlinearity of the inertial navigation and the 3D vision observation models [106, 122]. We propose a new formulation of the airborne 3D VSLAM based on a stereoscopic vision system and a nonlinear H∞ (NH∞) filtering scheme. In addition, this chapter provides robustness properties and consistency analysis of the Airborne VSLAM using EKF and robust $NH\infty$ filters. Simulation and experimental results validate the suitability of our approach.

5.3 Process Model

The localisation problem of an airborne system is formulated based on the core-sensing device, which is the Inertial Measurements Unit (IMU). This unit measures the acceleration *(ax, ay, az)* and the rotation rates *(p, q, r)* of the airborne platform with high

update rates. These quantities are then transformed and processed to provide the aerial vehicle position (X, Y, Z), velocity (U, V, W), and attitude (ϕ, θ, ψ) resulting in an Inertial Navigation System (INS).

Let us represent the INS with the following continues nonlinear model

$$\begin{cases} \dot{x} = f(x,u,w) \\ y = h(x,u,v) \end{cases} \quad \textbf{(5. 1)}$$

where $f(\cdot, \cdot, \cdot)$ is a non-linear state transition function which links the current state with the previous state and current control input. $h(\cdot, \cdot, \cdot)$ is a non-linear observation function, which links the observation to the current state.

x is the state vector, which contains the position in navigation frame, velocity in body frame and Euler angles, y is the observation vector and u represents the IMU outputs (angular rates, and accelerations) as follows:

$$x = [X, Y, Z, U, V, W, \phi, \theta, \psi]^T \quad \textbf{(5. 2)}$$

$$u = [p, q, r, ax, ay, az]^T \quad \textbf{(5. 3)}$$

w and v are respectively the process and observation noises.

The navigation equations require the definition of at least two reference frames, one frame for the body/inertial representation (vehicle) and another frame for the navigation representation. Then, the vehicle equations of motion are given by simple integration and frame transformations. Equations of motion of the vehicle are presented in detail in Section 2.3.2 of Chapter 2.

5.4 Observation Model

Cameras became an important sensor for aerial robotics because of their low cost and high quality of the acquired images, which are necessary for aerial mapping. Processing images using computer vision becomes vital for automatic perception and recognition of the environment. Stereoscopic vision is broadly defined as the recovery of three-dimensional characteristics of a scene from multiple images taken from two

different viewpoints. Stereo is an attractive source of information for machine perception because it leads to direct range measurements, and unlike monocular approaches, does not merely infer depth or orientation through the use of photometric and statistical assumptions. Once stereo images are brought into point-to-point correspondence, recovering range values is relatively straightforward. Another advantage is that stereo is a passive method. Although active ranging methods that use structured light, laser range finders, or other active sensing techniques are useful in tightly controlled domains such as industrial automation applications. However, they are clearly limited in use for some outdoor aerial vision problems.

5.4.1 Camera model

Perspective camera model includes intrinsic and extrinsic parameters. This model ensures the geometric transformation between camera/image and world/camera reference frames respectively Figure (5.1.a).

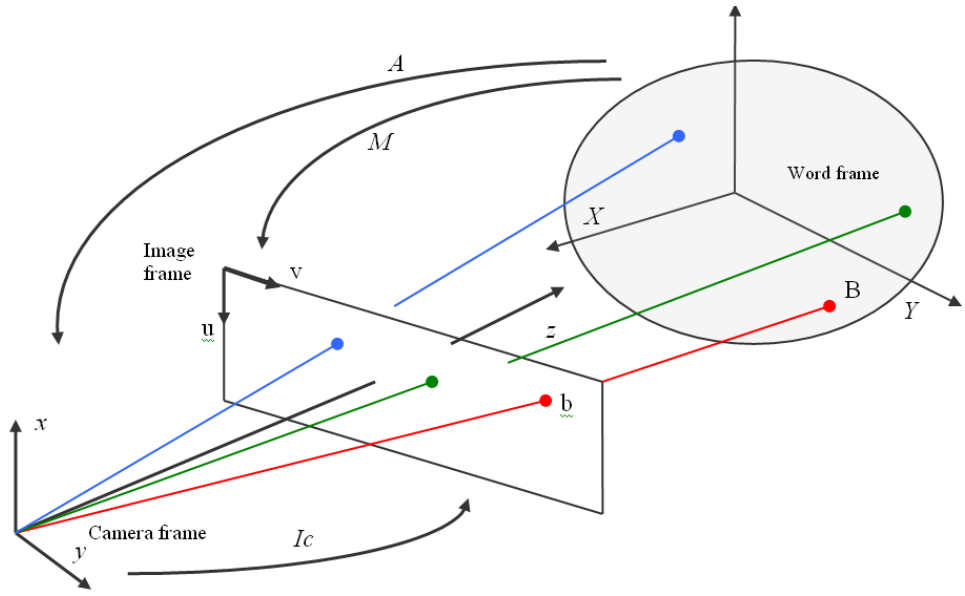


Figure.5.1. a Camera model

A.1 Intrinsic Parameters (transformation camera/image)

Intrinsic parameters of a camera are defined by the horizontal and vertical scale factor (k_v and k_u), the image centre coordinates ($u0, v0$) given in the image frame and the focal distance f as:

$$I_c = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \text{ with } \begin{cases} \alpha_u = -f \times k_u \\ \alpha_v = f \times k_v \end{cases}$$

A.2 Extrinsic Parameters (transformation world/camera)

These parameters define the transformation from the world to camera frame given by the homogeneity matrix A.

$$A = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \quad \textbf{(5.4)}$$

The matrix A is a combination of a rotation matrix R and a translation t from the world frame to the camera frame and obviously, the matrix A changes with the camera (UAV) displacement.

5.4.2 Airborne Stereo Vision

One of the important challenges for UAVs is to use low cost exteroceptive sensors to ensure a full autonomy. Stereovision cameras and computer vision algorithms become, nowadays, the most appropriate solution to reach this objective. Airborne stereo vision is more difficult than vision (Stereo or mono) for mobile ground robotics because of the 6 DoF requirements of the UAVs. In the following, we develop an observation model using stereoscopic cameras embedded on a UAV, Figure (5.1.b). Similar developments of camera observation models, mainly for mobile robotics applications, are presented in [92, 105], and [106].

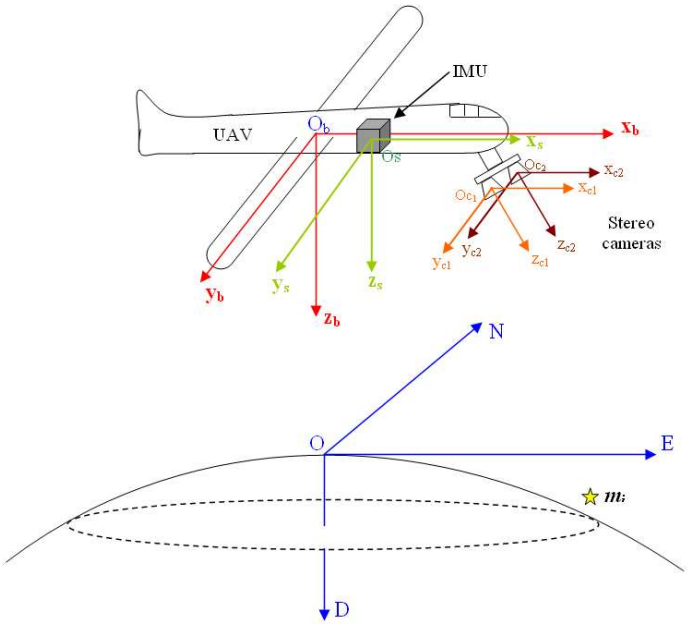


Figure.5.1. b UAV with IMU & stereo cameras

(O, N, E, D) is the navigation frame

(O_b, x_b, y_b, z_b) is the body frame

(O_s, x_s, y_s, z_s) is the IMU frame

$(O_{c1}, x_{c1}, y_{c1}, z_{c1})$ is the camera right frame

$(O_{c2}, x_{c2}, y_{c2}, z_{c2})$ is the camera left frame

The landmarks m_i represent specific feature points in the operation environment. These landmarks stand for features like corners or local extrema, which can be detected by feature extractors such as Harris, ASIFT and SIFT [81].

The landmark m_i coordinates in the navigation frame are given by:

$$\begin{aligned} \overrightarrow{Om_i}\big|_n &= \overrightarrow{OO_b} + \overrightarrow{O_bO_s} + \overrightarrow{O_sO_{c1}} + \overrightarrow{O_{c1}m_i} \\ \overrightarrow{Om_i}\big|_n &= \overrightarrow{OO_b} + \overrightarrow{O_bO_s} + \overrightarrow{O_sO_{c2}} + \overrightarrow{O_{c2}m_i} \end{aligned} \qquad \textbf{(5.5)}$$

B.1 Transformation Body (UAV)/navigation frame:

$\overrightarrow{OO_b} = \begin{bmatrix} X_{uav} & Y_{uav} & Z_{uav} \end{bmatrix}^T$ is the position of the UAV in the navigation frame. This position is given by the INS.

B.2 Transformation IMU/ Body (UAV) frame:

$$\overrightarrow{O_bO_s}\Big|_n = C_b^n.\overrightarrow{O_bO_s}\Big|_b = C_b^n.\begin{bmatrix} x_s & y_s & z_s \end{bmatrix}^T \quad \textbf{(5.6)}$$

where C_b^n is a rotation matrix that transforms vectors from body to navigation frame. $\begin{bmatrix} x_s & y_s & z_s \end{bmatrix}^T$ is the position of the IMU presented in body frame. If the IMU is at the centre of gravity then $\begin{bmatrix} x_s & y_s & z_s \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$.

B.3 Transformation IMU/cameras frames:

$$\overrightarrow{O_sO_{c1}}\Big|_n = C_b^n \cdot C_s^b \cdot \begin{bmatrix} x_{c1} & y_{c1} & z_{c1} \end{bmatrix}^T \quad \textbf{(5.7)}$$

$$\overrightarrow{O_sO_{c2}}\Big|_n = C_b^n \cdot C_s^b \cdot \begin{bmatrix} x_{c2} & y_{c2} & z_{c2} \end{bmatrix}^T \quad \textbf{(5.8)}$$

where C_s^b is a rotation matrix that transforms vectors from IMU to body frame. $\begin{bmatrix} x_{c1} & y_{c1} & z_{c1} \end{bmatrix}^T$ ($\begin{bmatrix} x_{c2} & y_{c2} & z_{c2} \end{bmatrix}^T$) is the position of the camera right (left) in the IMU frame presented in IMU frame.

B.4 Landmark coordinates in cameras frames

$$\overrightarrow{O_{c1}m_i}\Big|_n = C_b^n \cdot C_s^b \cdot C_{c1}^s \cdot [x_{mi} \quad y_{mi} \quad z_{mi}]_{c1}^T = M_{c1}^n \cdot [x_{mi} \quad y_{mi} \quad z_{mi}]_{c1}^T \quad \textbf{(5.9)}$$

$$\overrightarrow{O_{c2}m_i}\Big|_n = C_b^n \cdot C_s^b \cdot C_{c2}^s \cdot [x_{mi} \quad y_{mi} \quad z_{mi}]_{c2}^T = M_{c2}^n \cdot [x_{mi} \quad y_{mi} \quad z_{mi}]_{c2}^T \quad \textbf{(5.10)}$$

where C_{c1}^s (C_{c2}^s) is a rotation matrix that transforms vectors from the right (left) camera to the IMU frame. $\begin{bmatrix} x_{m1} & y_{m1} & z_{m1} \end{bmatrix}^T$ ($\begin{bmatrix} x_{m2} & y_{m2} & z_{m2} \end{bmatrix}^T$) is the position of the landmark m_i in the right (left) camera presented in right (left) camera frame.

B.5 Landmark coordinates in navigation frame

$$\begin{aligned} \overrightarrow{Om_i}\Big|_n &= \begin{bmatrix} X_{uav} & Y_{uav} & Z_{uav} \end{bmatrix}_n^T + C_b^n.\begin{bmatrix} x_s & y_s & z_s \end{bmatrix}_b^T + C_b^n \cdot C_s^b \cdot \begin{bmatrix} x_{c1} & y_{c1} & z_{c1} \end{bmatrix}_s^T + M_{c1}^n \cdot [x_{mi} \quad y_{mi} \quad z_{mi}]_{c1}^T \\ \overrightarrow{Om_i}\Big|_n &= \begin{bmatrix} X_{uav} & Y_{uav} & Z_{uav} \end{bmatrix}_n^T + C_b^n.\begin{bmatrix} x_s & y_s & z_s \end{bmatrix}_b^T + C_b^n \cdot C_s^b \cdot \begin{bmatrix} x_{c2} & y_{c2} & z_{c2} \end{bmatrix}_s^T + M_{c2}^n \cdot [x_{mi} \quad y_{mi} \quad z_{mi}]_{c2}^T \end{aligned} \quad \textbf{(5.11)}$$

If we put

$$\begin{aligned}\overrightarrow{OO_{c1}}\Big|_n &= T_{c1} = \begin{bmatrix} X_{uav} & Y_{uav} & Z_{uav} \end{bmatrix}_n^T + C_b^n . \begin{bmatrix} x_s & y_s & z_s \end{bmatrix}_b^T + C_b^n \cdot C_s^b \cdot \begin{bmatrix} x_{c1} & y_{c1} & z_{c1} \end{bmatrix}_s^T \\ \overrightarrow{OO_{c2}}\Big|_n &= T_{c2} = \begin{bmatrix} X_{uav} & Y_{uav} & Z_{uav} \end{bmatrix}_n^T + C_b^n . \begin{bmatrix} x_s & y_s & z_s \end{bmatrix}_b^T + C_b^n \cdot C_s^b \cdot \begin{bmatrix} x_{c2} & y_{c2} & z_{c2} \end{bmatrix}_s^T \end{aligned} \quad \textbf{(5.12)}$$

Then

$$\overrightarrow{Omi}\Big|_n = \begin{bmatrix} M_{c1}^n & T_{c1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{mi} & y_{mi} & z_{mi} & 1 \end{bmatrix}_{c1}^T = \begin{bmatrix} M_{c2}^n & T_{c2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{mi} & y_{mi} & z_{mi} & 1 \end{bmatrix}_{c2}^T \quad \textbf{(5.13)}$$

$$\overrightarrow{Omi}\Big|_n = Mh_{c1}^n \cdot \begin{bmatrix} x_{mi} & y_{mi} & z_{mi} & 1 \end{bmatrix}_{c1}^T = Mh_{c2}^n \cdot \begin{bmatrix} x_{mi} & y_{mi} & z_{mi} & 1 \end{bmatrix}_{c2}^T \quad \textbf{(5.14)}$$

B.6 Transformation camera/image frame

Using the cameras model (transformation camera/image), we can conclude:

$$\begin{aligned} \begin{bmatrix} s_1 u_1 & s_1 v_1 & s_1 \end{bmatrix}^T &= I_{c1} \cdot \begin{bmatrix} x_{mi} & y_{mi} & z_{mi} & 1 \end{bmatrix}_{c1}^T \\ \begin{bmatrix} s_2 u_2 & s_2 v_2 & s_2 \end{bmatrix}^T &= I_{c2} \cdot \begin{bmatrix} x_{mi} & y_{mi} & z_{mi} & 1 \end{bmatrix}_{c2}^T \end{aligned} \quad \textbf{(5.15)}$$

I_{c1} and I_{c2} are the matrix of intrinsic parameters of the right and the left camera respectively (Section A.1)

Then

$$\begin{bmatrix} s_1 u_1 & s_1 v_1 & s_1 \end{bmatrix}^T = I_{c1} \cdot (Mh_{c1}^n)^{-1} \cdot \overrightarrow{Om_i}\Big|_n \quad \textbf{(5.16)}$$

$$\begin{bmatrix} s_2 u_2 & s_2 v_2 & s_2 \end{bmatrix}^T = I_{c2} \cdot (Mh_{c2}^n)^{-1} \cdot \overrightarrow{Om_i}\Big|_n \quad \textbf{(5.17)}$$

After development, the scale factors, for camera right and left s_1 and s_2 respectively, are given by the following equation:

$$s_1 = m_{31}^{c1} x_{mi}^n + m_{32}^{c1} y_{mi}^n + m_{33}^{c1} z_{mi}^n + m_{34}^{c1} \quad \textbf{(5.18)}$$

$$s_2 = m_{31}^{c2} x_{mi}^n + m_{32}^{c2} y_{mi}^n + m_{33}^{c2} z_{mi}^n + m_{34}^{c2} \quad \textbf{(5.19)}$$

Therefore, the **observation model,** linking the perceived visual landmarks to the SLAM state vector is given by:

$$\begin{cases} u_1 = \dfrac{m_{11}^{c1}x_{mi}^n + m_{12}^{c1}y_{mi}^n + m_{13}^{c1}z_{mi}^n + m_{14}^{c1}}{m_{31}^{c1}x_{mi}^n + m_{32}^{c1}y_{mi}^n + m_{33}^{c1}z_{mi}^n + m_{34}^{c1}} \\ v_1 = \dfrac{m_{21}^{c1}x_{mi}^n + m_{22}^{c1}y_{mi}^n + m_{23}^{c1}z_{mi}^n + m_{24}^{c1}}{m_{31}^{c1}x_{mi}^n + m_{32}^{c1}y_{mi}^n + m_{33}^{c1}z_{mi}^n + m_{34}^{c1}} \\ u_2 = \dfrac{m_{11}^{c2}x_{mi}^n + m_{12}^{c2}y_{mi}^n + m_{13}^{c2}z_{mi}^n + m_{14}^{c2}}{m_{31}^{c2}x_{mi}^n + m_{32}^{c2}y_{mi}^n + m_{33}^{c2}z_{mi}^n + m_{34}^{c2}} \\ v_2 = \dfrac{m_{21}^{c2}x_{mi}^n + m_{22}^{c2}y_{mi}^n + m_{23}^{c2}z_{mi}^n + m_{24}^{c2}}{m_{31}^{c2}x_{mi}^n + m_{32}^{c2}y_{mi}^n + m_{33}^{c2}z_{mi}^n + m_{34}^{c2}} \end{cases} \quad \textbf{(5.20)}$$

where $[x_{mi}^n \quad y_{mi}^n \quad z_{mi}^n]^T$ is the coordinate of the landmark m_i in the navigation frame (NED). m_{ij}^{c1} and m_{ij}^{c2} are the components of $I_{c1} \cdot (Mh_{c1}^n)^{-1}$ and $I_{c2} \cdot (Mh_{c2}^n)^{-1}$ respectively.

5.5 EKF Airborne VSLAM

Kalman Filter (KF) is an effective stochastic estimator for a large number of problems. However, as in most real applications, the process and/or observation models are nonlinear therefore the linear Kalman filter algorithm cannot be directly applied. To overcome this, an Extended Kalman Filter (EKF) is proposed. Based on this technique, the navigation solution uses the current estimated states from the filter as the linearisation reference from which the estimation procedure can start at each time step *k*. If the filter operates properly, the linearisation error around the estimated solution can be maintained reasonably small. However, if the filter is ill-conditioned due to modelling errors, incorrect tuning of the covariance matrix, or initialisation error, then the estimation error will affect the linearisation error, which in turn will affect the estimation process. This is known as a filter divergence. For this reason, the EKF requires greater care in modelling and tuning than the linear Kalman filter.

5.5.1 EKF Filter

The state vector of the UAV SLAM is given by:

$$x = [x_v \; x_m]$$

$$x_v = [X, Y, Z, U, V, W, \phi, \theta, \psi]^T, \; x_m = [m_1 \; m_2 \; m_3 ... m_N]$$

x_v is the state vector of the vehicle and x_m is the state vector of the landmarks observed during the UAV flight. Landmark initialisation algorithm is based on triangulation methods, which directly calculates a three-dimensional position of landmark based on stereovision measurements and the Inverse Model of Observation (IMO).

An augmented version of Equation (5.1) can be written as a non-linear discrete time state transition equation:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}) + g(x_{k-1}) w_{k-1} \\ y_k &= h(x_k, v_k) \end{aligned} \quad \textbf{(5. 21)}$$

where f is the discrete version of Equation (2.9) (in addition to elements of the landmarks states), g is a nonlinear function, x_k is the state at time step k, w_k is some additive process noises, y_k is the observation made at time k, v_k is some additive observation noises. We assume that w_k and v_k are uncorrelated zero mean Gaussian with known covariance Q_k and R_k. The objective of the filtering technique is, then, to estimate x_k using available observation y_k. Predictor-Corrector formulation of the EKF is presented in detail in Section 3.4 of Chapter 3, Equations (3.19-3.23). Where $P_{k/k}$ is the estimated error covariance for the system:

$$P_{k/k} = \begin{bmatrix} P_{k/k}^{vv} & P_{k/k}^{v1} & \cdots & P_{k/k}^{vN} \\ P_{k/k}^{vv} & P_{k/k}^{11} & \cdots & P_{k/k}^{1N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k/k}^{Nv} & P_{k/k}^{N1} & \cdots & P_{k/k}^{NN} \end{bmatrix} \quad \textbf{(5. 22)}$$

with the sub-matrices $P_{k/k}^{vv}$, $P_{k/k}^{vi}$, $P_{k/k}^{ii}$ $i = 1,...N$ are the UAV-to-UAV, UAV-to-landmark and landmark-to-landmark covariances respectively.

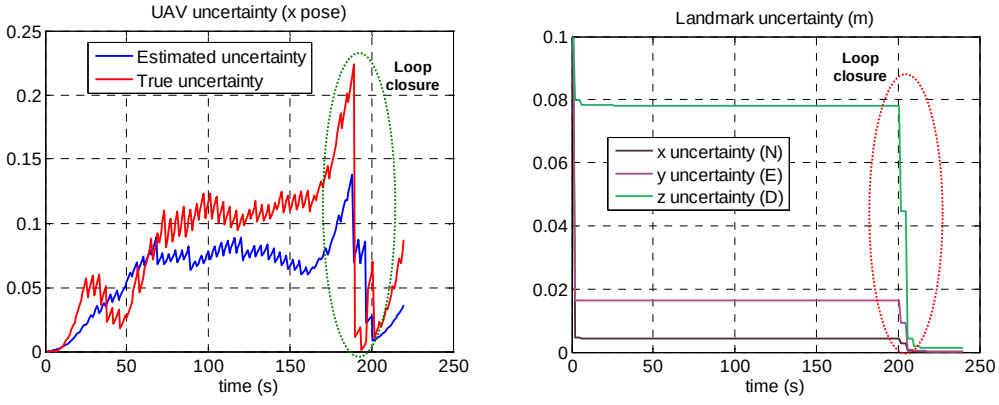
5.5.2 Loop Closing

As the UAV moves in the environment, it builds a map of the landmarks and correlates this information in order to determine precisely the UAV location. One solution to the correlation problem was introduced by Smith, Self and Cheeseman [107], and is called the Stochastic Map (SM). The SM allows for the concurrent

mapping of landmarks and localisation of the UAV with respect to the landmarks using EKF architecture. The Stochastic Map is essentially an augmented EKF, where the observed landmarks are stored in the filter state vector along with the vehicle states.

Loop closing means the capacity of recognising previously detected landmarks to reduce the uncertainties in both the UAV and the landmark positions. Using an augmented state vector where information about the landmarks is stored, the loop closure detection becomes an automatic task.

Consider the scenario of a UAV navigating and making a map loop. The curves shown in Figure (5.2) represent the evolution of uncertainty for the UAV (x-position), and one observed landmark "*m*". This landmark is observed at the beginning of UAV navigation and is observed again after ***200s*** of navigation. From Figure (5.2) we observe the reduction of the uncertainties, both, for the UAV, Figure (5.2.a) and the landmark, Figure (5.2.b) positions when the loop closing is detected at ***t=200s***. At this time, the UAV detects and recognises some landmarks observed previously at ***t=0s***. The loop closing detection leads to a significant decrease in the true and estimated uncertainty for localisation and mapping. However, the main problem of the augmented map method (SM), which facilitates the loop closing detection, is the complexity that increases exponentially with the number of observed landmarks. This problem will be solved in Section 5.8, when a new approach of map management is proposed.



-a- -b-

Figure.5. 2 Loop closure detection at t=200s, a) UAV uncertainty, b) Landmark uncertainty,

5.6 Airborne VSLAM Observability

Observability of nonlinear systems is not very easy to analyse. Many papers treat the SLAM problem without mentioning or examining the observability of the system. Observability, if it does not hold, can lead to an inconsistent SLAM solution [108-109]. Only very few papers have tried to treat the observability of VSLAM, mainly the local observability, which constitutes only a partial solution of the problem. The best way to prove the global observability of nonlinear systems is to use Lie derivatives [104]. However, as it is very hard to develop this method in our Airborne VSLAM case, where the size of state vector is larger than 12 (9 states for the UAV, 3 states for each landmark), we propose to analyse the observability problem with a reduced SLAM system assuming a planar displacement of the UAV (at a fixed altitude). The cameras are heading down vertically. This leads to a three DoF system with six model states (one landmark), rather than 12 states, as follows:

$$\dot{x} = f(x,u) = \begin{bmatrix} U\cos(\psi) - V\sin(\psi) \\ U\sin(\psi) + V\cos(\psi) \\ \psi \\ 0 \\ 0 \\ 0 \end{bmatrix}, x = \begin{bmatrix} X_{uav} & Y_{uav} & \psi & x_{mi} & y_{mi} & z_{mi} \end{bmatrix}^T, \; x \in \Re^6 \qquad \textbf{(5. 23)}$$

where $[x_{mi}, y_{mi}, z_{mi}]^T$ are the coordinates of the landmark in navigation frame. $\alpha_{ui}, \alpha_{vi}, u_{0i}, v_{0i}$, $i \in \{1,2\}$ are the intrinsic parameters of the right and left cameras.

$$y = h(x) = \begin{bmatrix} \frac{\alpha_{u1}(\cos(\psi)x_{mi} + \sin(\psi)y_{mi} - \sin(\psi)Y_{uav} - \cos(\psi)X_{uav})}{z_{mi}} + u_{01} \\ \frac{\alpha_{v1}(-\sin(\psi)x_{mi} + \sin(\psi)X_{uav} - y_{c1} + \cos(\psi)y_{mi} - \cos(\psi)Y_{uav})}{z_{mi}} + v_{01} \\ \frac{\alpha_{u2}(\cos(\psi)x_{mi} + \sin(\psi)y_{mi} - \sin(\psi)Y_{uav} - \cos(\psi)X_{uav})}{z_{mi}} + u_{02} \\ \frac{\alpha_{v2}(-\sin(\psi)x_{mi} + \sin(\psi)X_{uav} - y_{c1} + \cos(\psi)y_{mi} - \cos(\psi)Y_{uav})}{z_{mi}} + v_{02} \end{bmatrix}, \qquad y \in \Re^4$$

This system satisfies the observability rank condition if any of the observability matrices are of rank 6 (recall that $x \in \Re^6$), where the observability matrix is given by the Lie derivatives as:

$$O = \frac{\partial}{\partial x_k}[Lf_0^T \quad Lf_1^T \quad Lf_2^T \quad Lf_3^T \quad Lf_4^T \quad Lf_5^T]^T \tag{5. 24}$$

with

$$\begin{aligned} Lf_0 &= h(x) = y \\ Lf_i &= \frac{\partial Lf_{i-1}}{\partial x_k} x_{k+1} \end{aligned} \tag{5. 25}$$

We find that $rank(O) = 4 < 6$, which means that the SLAM is not observable. This can be explained by the fact that camera observations still relative measures and cannot estimate an absolute position in the global frame (landmarks uncertainties and UAV position uncertainty are highly dependent). One solution of this limitation is to introduce a known landmark $m_* = [x_{known}, y_{known}\ z_{known}]^T$ to the state vector. Then, the observation model becomes:

$$y' = h'(x) = \begin{bmatrix} \frac{\alpha_{u1}(\cos(\psi)x_{mi} + \sin(\psi)y_{mi} - \sin(\psi)Y_{uav} - \cos(\psi)X_{uav})}{z_{mi}} + u_{01} \\ \frac{\alpha_{v1}(-\sin(\psi)x_{mi} + \sin(\psi)X_{uav} - y_{c1} + \cos(\psi)y_{mi} - \cos(\psi)Y_{uav})}{z_{mi}} + v_{01} \\ \frac{\alpha_{u2}(\cos(\psi)x_{mi} + \sin(\psi)y_{mi} - \sin(\psi)Y_{uav} - \cos(\psi)X_{uav})}{z_{mi}} + u_{02} \\ \frac{\alpha_{v2}(-\sin(\psi)x_{mi} + \sin(\psi)X_{uav} - y_{c1} + \cos(\psi)y_{mi} - \cos(\psi)Y_{uav})}{z_{mi}} + v_{02} \\ \frac{\alpha_{u1}(\cos(\psi)x_{known} + \sin(\psi)y_{known} - \sin(\psi)Y_{uav} - \cos(\psi)X_{uav})}{z_{known}} + u_{01} \\ \frac{\alpha_{v1}(-\sin(\psi)x_{known} + \sin(\psi)X_{uav} - y_{c1} + \cos(\psi)y_{known} - \cos(\psi)Y_{uav})}{z_{known}} + v_{01} \\ \frac{\alpha_{u2}(\cos(\psi)x_{known} + \sin(\psi)y_{known} - \sin(\psi)Y_{uav} - \cos(\psi)X_{uav})}{z_{known}} + u_{02} \\ \frac{\alpha_{v2}(-\sin(\psi)x_{known} + \sin(\psi)X_{uav} - y_{c1} + \cos(\psi)y_{known} - \cos(\psi)Y_{uav})}{z_{known}} + v_{02} \end{bmatrix} \tag{5. 26}$$

In this case we find that $rank(O) = 6$, which means that the SLAM problem becomes observable and thus consistent. However a known feature is not a practical solution especially in outdoor environment. Thus, proposing an observable and robust airborne VSLAM algorithm presents another contribution we give in the following section.

5.7 Robust Observable VSLAM

5.7.1 EKF VSLAM Consistency

Lower bound for the map accuracy based on EKF SLAM is violated and produces inconsistent estimates (especially UAV position) [90, 102]. Shoudong, in his paper [104], provides both convergence properties and a consistency analysis for some basic scenarios of the nonlinear two-dimensional EKF SLAM problem. A number of recent publications [102, 103] indicate that the key source of EKF SLAM inconsistency is the error introduced during the linearisation process. While it is clear that linearisation is an approximation that can introduce errors into the estimation process it is reasonable to expect that the incorrect estimate is likely to be either too optimistic (the estimated uncertainty smaller than the true uncertainty) or too pessimistic (the estimated uncertainty larger than the true uncertainty). In the literature related to SLAM, only estimator inconsistency, as a result of optimistic estimates, is reported [104].

Furthermore, the EKF SLAM requires an accurate enough process model and known process and observation noise characteristics (centred Gaussian noises). Moreover, in all the theoretical convergence properties proved in the previous works, [90, 104], it is assumed that the Jacobians are evaluated at the true UAV pose and the true landmark positions. For realistic SLAM scenarios, the true locations of the UAV and landmarks are not known, and the Jacobians have to be evaluated at the estimated values. This may result in overconfident (inconsistent) estimates [104].

In this section, we propose a solution dealing with robustness and consistency problems for the Airborne VSLAM by introducing the nonlinear $H\infty$ (*NH∞)* filter scheme presented in details in Section 3.6, Chapter 3.

5.7.2 Nonlinear $H\infty$ Filter

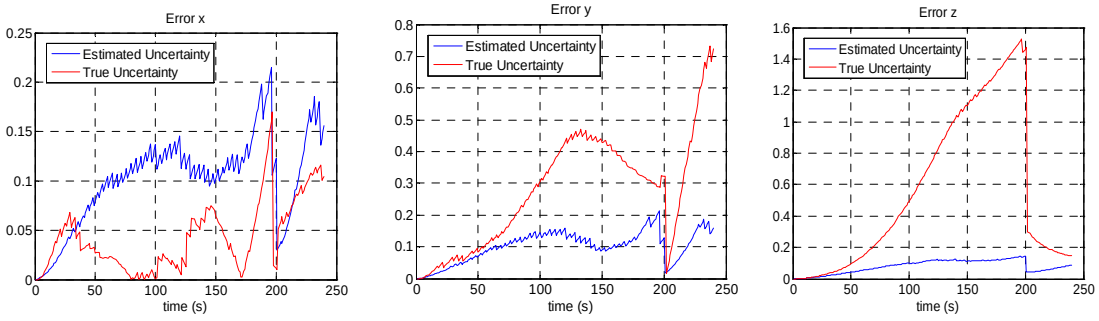
Nonlinear $H\infty$ filter could handle modelling errors and noise uncertainties while minimising the worst-case estimation error rather than the covariance of the estimation error. State estimators that can tolerate such uncertainties are called robust. Although robust estimators based on Kalman filter theory can be designed, these approaches are somewhat ad-hoc in that they attempt to modify an already existing approach. In contrast, the $H\infty$ filter was specifically designed for optimality and robustness [66, 68].

The NH∞ filter principle and algorithm are presented in details in Section 3.6 of Chapter 3.

5.7.3 Consistent EKF (NH∞) Airborne VSLAM

Even if the use of a robust filter, such as NH∞ filter, improves the performance of the airborne VSLAM algorithm (Optimality and Robustness) and increases the consistency, the problem of estimating the UAV states in an absolute frame using relative and uncertain data, is still posed. This problem is directly linked with the above mentioned issue of 'full observability' without known features (Section 5.6) [110-111].

The following figures show an example of the EKF VSLAM inconsistency based on the UAV position errors along the axes ***x, y*** and ***z***. It is clear, from Figure (5.3), that the estimated uncertainty is very small compared to the true uncertainty specifically for the UAV ***y*** and ***z*** states. At ***t=200s,*** we observe the increase of the VSLAM consistency because of the loop closure detection Figure (5.3. a, b and c).



-a- -b- -c-

Figure.5. 3 Estimated and True uncertainty for the UAV position

As a solution, we propose the following strategy to improve the consistency of the VSLAM filtering scheme whether it is using EKF filter or *NH*∞ filter:

Let us assume that at ***t=0,*** the UAV position is known accurately in an absolute frame. From this position, the UAV observes landmarks with their associated uncertainties, which depend on the camera's uncertainties. If this observation is processed ***n*** times, through the filtering scheme (EKF), then the landmark's position uncertainties decrease to a lower bound, Figure (5.4) and Figure (5.5). Once this is achieved, these landmarks can be considered as a relative known feature, which ensure the full observability of the filter, Equation (5.26).

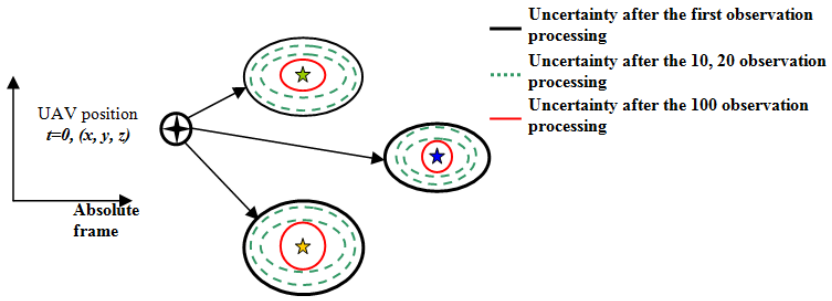
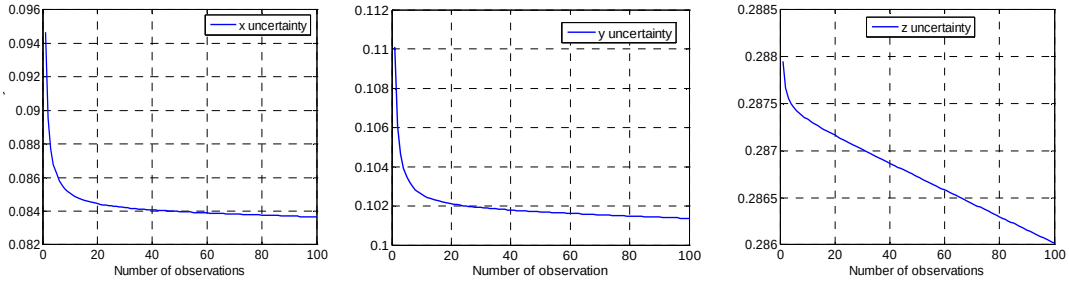


Figure.5. 4 Evolution of landmark uncertainties with the number of observation processing



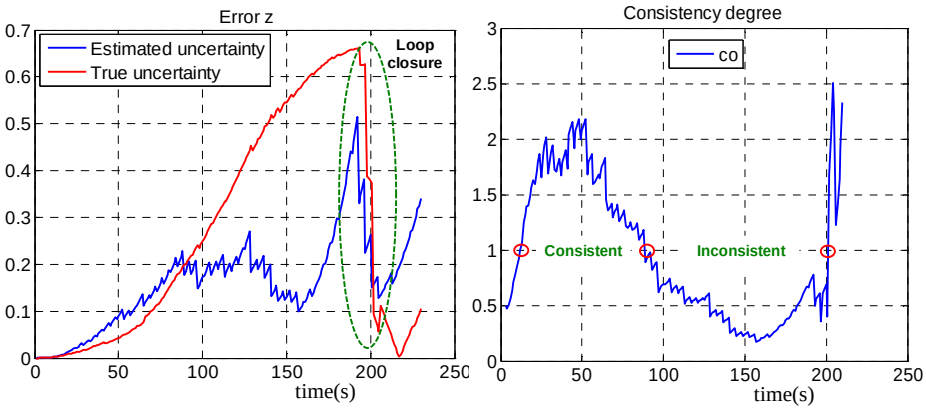
x-uncertainty y-uncertainty z-uncertainty

Figure.5. 5 Landmark uncertainties after 100 observations processed

To evaluate the degree of SLAM consistency, we define the parameter *co* as:

$$co = \frac{Estimated\ \ uncertainty}{True\ uncertainty} \text{ and propose: } \begin{cases} co > 1 => consistent\ VSLAM \\ co \le 1 => inconsistent\ VSLAM \end{cases}$$

This parameter, calculated at each sampling time of simulation, tells us about the instantaneous VSLAM consistency. However, this is not enough to inform us about the true consistency of the VSLAM filtering scheme (EKF is used here). As shown in Figure (5.6.a), just after the loop closure is detected at $t=200s, co = 0.18/0.08 = 2.25 > 1$ which signifies the EKF VSLAM is consistent. On the other hand, it is clear, from the same figure, that for $t \in [90s..200s]$, the estimated uncertainty is smaller than the true uncertainty, which signifies the EKF SLAM inconsistency [104]. These results are confirmed by the consistency degree *co* given by Figure (5.6.b).



a) Estimated and True *z* uncertainty b) Evolution of the consistency degree

Figure.5. 6 Consistency analysis

To avoid this kind of erroneous analysis, we propose to check the SLAM (based on EKF here) consistency in the worst-case situation, which is evaluated by the robust consistency parameter Rco as: $Rco = \min\left(\frac{Estimated\ \ uncertainty}{True\ uncertainty}\right)$. This latter is calculated at the last of the simulation telling us about the whole VSLAM consistency. Using this parameter for the simulation in Figure (5.6.b), the minimal value of co obtained at $t=160s$ is $Rco = 0.22 < 1$, which shows clearly the EKF SLAM inconsistency.

In the following experiment, the UAV navigates and makes landmark observations. Each observed landmark is used in **one (ten)** observation processing (update). In other words, Equation (3.21) for EKF or Equation (3.47) for NH∞ update is processed **one (ten)** time(s) at each filter sampling time. The following tables show the values of Rco when we used one and ten updates (observation processing) of the landmark observations.

As can be seen from the tables, the estimation accuracy increases when the number of observation processing increases. Thus, we can conclude that by observing the landmarks and processing them ***n*** times, we increase the consistency of the filter (Table 5.1 and Table 5.2). Our approach will increase, a little, the airborne VSLAM algorithm complexity. However, this complexity is primarily dependent on the choice of landmarks and their initial observation uncertainties.

state	One Observation Processing (update)						
	Estimated uncertainty			True uncertainty			*Rco*
	min	max	mean	min	max	mean	
x	0	0.1251	0.0722	0	0.0619	0.0173	0.9033
y	0	0.1463	0.0726	0	0.2812	0.1113	0.0146
z	0	0.0995	0.0457	0	0.4908	0.1497	0.0003
U	0.0042	0.0673	0.0464	0	0.0546	0.0248	0.0161
V	0.0042	0.0767	0.0469	0	0.0854	0.0575	1.8452
W	0.0022	0.0258	0.0165	0	0.1080	0.0531	8.6046
θ	0.0004	0.0028	0.0022	0	0.0014	0.0008	7.8261
ϕ	0.0004	0.0027	0.0022	0	0.0048	0.0029	191.8137
ψ	0.0004	0.0038	0.0027	0	0.0041	0.0021	15.8661

Table 5. 1 Degree of consistency with one observation processing

state	10 Observation Processed (updates)						
	Estimated uncertainty			True uncertainty			*Rco*
	min	max	mean	min	max	mean	
x	0	0.104	0.0578	0	0.0297	0.0105	1.5416
y	0	0.128	0.0566	0	0.1097	0.0425	0.0422
z	0	0.0901	0.0406	0	0.2933	0.0724	0.0369
U	0.0042	0.0507	0.0326	0	0.0362	0.0165	0.4091
V	0.0042	0.0659	0.0382	0	0.0783	0.0532	3.8507
W	0.0022	0.0242	0.0157	0	0.0809	0.039	0.0455
ϕ	0.0004	0.0027	0.0021	0	0.0035	0.0013	9.5392
θ	0.0004	0.0025	0.0021	0	0.0043	0.0033	29.0489
ψ	0.0004	0.0034	0.0025	0	0.0027	0.0011	7.5769

Table 5. 2 Degree of consistency with ten observations processed

The solution proposed in this sub-section will be used later for the EKF based Airborne VSLAM and the NH∞ based Airborne VSLAM as mean of increasing the consistency of the proposed VSLAM solution.

5.8 3D VSLAM Map Management

A critical issue in a SLAM algorithm is to decide how to represent the joint distribution over vehicle poses and feature map. In particular, the map management is very important due to the fact that maps are usually represented by a high number of parameters [90], [105], [106] and [111].

The $O(N^2)$ complexity of the Kalman filter (N being the dimension of the state vector) does not allow large environments to be efficiently mapped since it limits the total number of landmarks that can be stored in the map [90]. Beyond this upper limit, real-time processing is no longer possible [105]. To prevent the state vector from a rapid dimension increase that would dramatically limit the mapping capacity of our VSLAM system, we introduce a map management approach based on a landmark performance criterion, which allows the substitution of some elements of the state vector. The proposed approach is very efficient and is based on the following simple to implement concept:

At each time step, the state vector is proposed to contain the new observed features and the best k-landmarks observed previously as illustrated in Figure (5.7). It is clear that the size of the state vector at each time is limited to (***k + number of new landmarks)*** (in our experiment ***k=20***). The same limitation is used for the covariance matrix. In order to maintain the correlation between landmarks and UAV/landmarks, we propose to save all the observed landmarks, during the flight, with their respective covariance matrices. This will help in building a large map and to make the loop closing detection possible.

- **K-landmarks performance criteria**

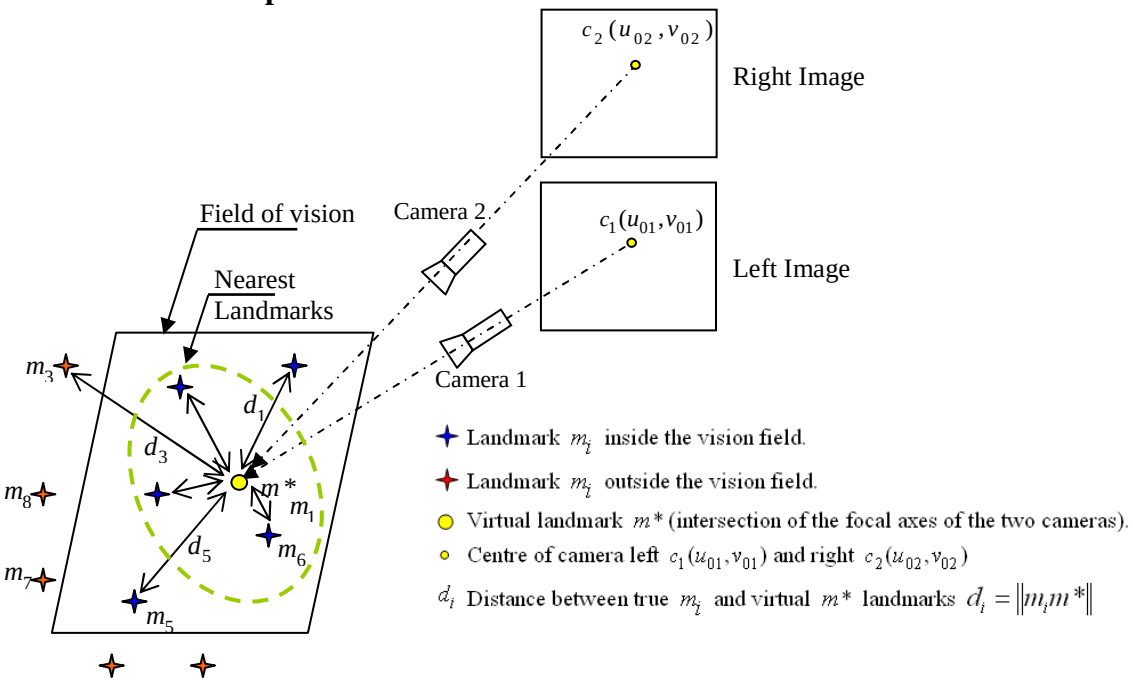


Figure.5. 7 Map management using the nearest k-landmarks approach

As can be seen from Figure (5.7), our approach is based on measuring the Euclidean distance of observed landmarks m_i and a virtual landmark $m*$. The latter is given by the intersection of the focal axes of the two cameras making it the virtual centre point of the real image. The best ***k-landmarks*** are the nearest ***k-landmarks*** to that virtual point. Subsequently, at each time step the state vector will contain the new observed landmarks and the ***k***, previously observed, nearest landmarks to the virtual centre point. The proposed map management approach reduces the size of the state vector significantly; furthermore, it makes the loop closing detection possible.

5.9 Simulation Results and Discussions

In the following simulations, we present a number of UAV flight scenarios for which NH∞ and EKF Airborne VSLAM schemes are developed and compared. The proposed map management approach is also assessed against the classical stochastic map approach.

In our simulation we assume a UAV navigating in unknown environment using IMU and stereo cameras models. At each step time a set of features will be observed.

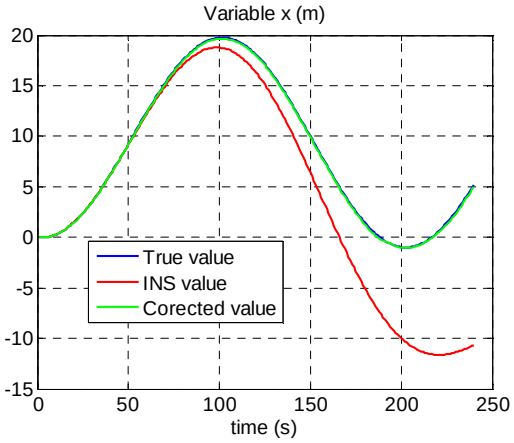
The observed features are divided into two classes: re-observed features are used to update the map and UAV pose, and new features are initialised and added to the map using the Inverse Observation Model (IOM).

5.9.1 EKF Airborne VSLAM

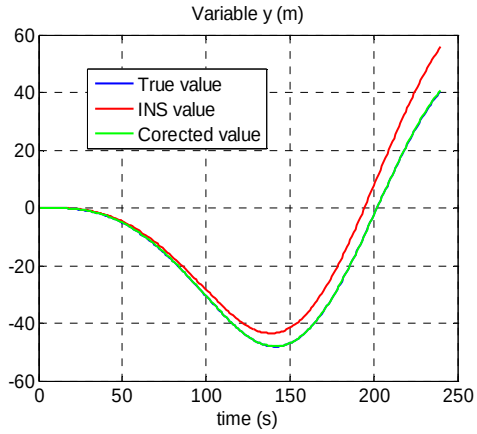
As can be seen from Figure (5.8. a, b, c and d), when the noise is a centred Gaussian, reasonably good EKF based estimation are obtained for ***x***, ***y*** and ***z***, presenting much better estimated trajectory than the INS trajectory. The inaccuracy of this estimator in z, Figure (5.8.c), affects the performance of the UAV trajectory estimation as shown in Figure (5.8.d). This last figure shows a scenario of a UAV making a loop. Although the corrected position is much better than the INS position, it still diverges from the true position up to (t=***200s***). When a loop-closure is detected, at this time a significant accuracy is observed and the corrected position is confused with the true position. The loop closing effect appears obviously in Figure (5.8.c) (dashed ellipse).

When, the process and observation noises are non Gaussian, the performance of the EKF estimator decreases significantly as shown in Figure (5.9). In this case, the corrected values of ***x*** and ***y*** are still very close to their corresponding true values. This can be explained by the efficiency of the stereo cameras to observe the bearing information nevertheless the range information (***z***) is given with less accuracy as in Figure (5.9.c). As shown in this figure, the value of ***z*** diverges, as the INS information, before it starts slightly converging to the true value at ***t=150s***. At ***t=200s***, which corresponds to the detection of the loop closing, we observe the full convergence of the corrected values of ***z*** to their corresponding true values.

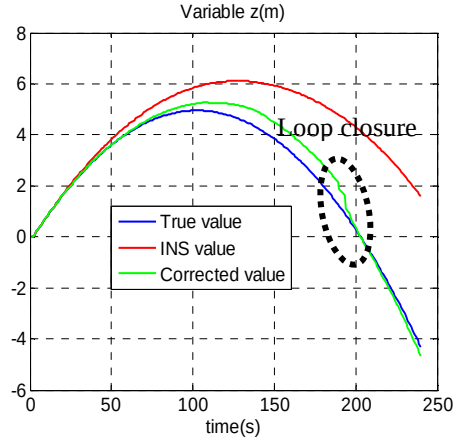
The problem of the EKF VSLAM is the poor consistency caused by the linearisation problem and the assumed characteristics of the process and observation noises, which should be centred Gaussian noises.



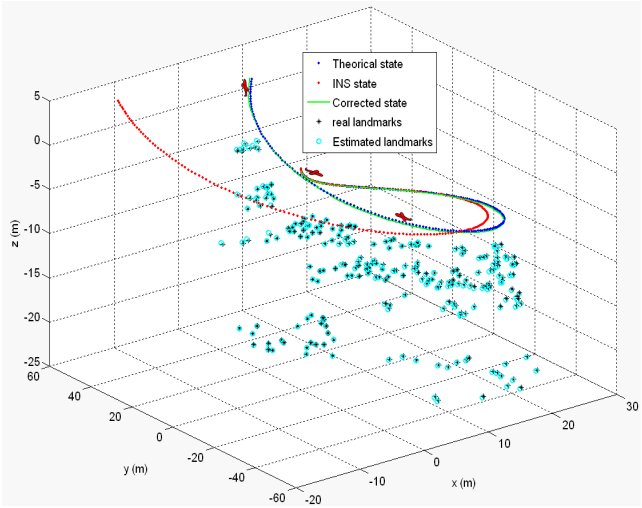
-a-



-b-

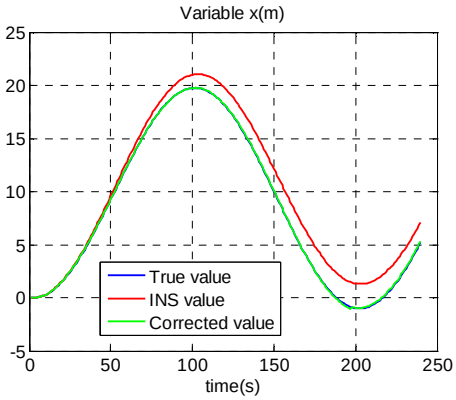


-c-

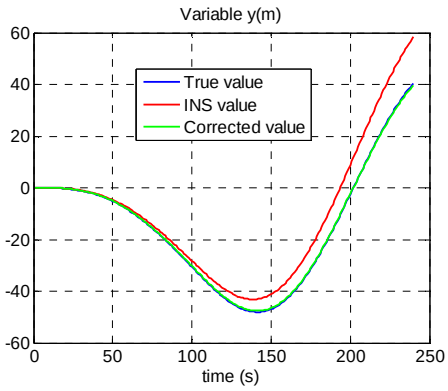


-d-

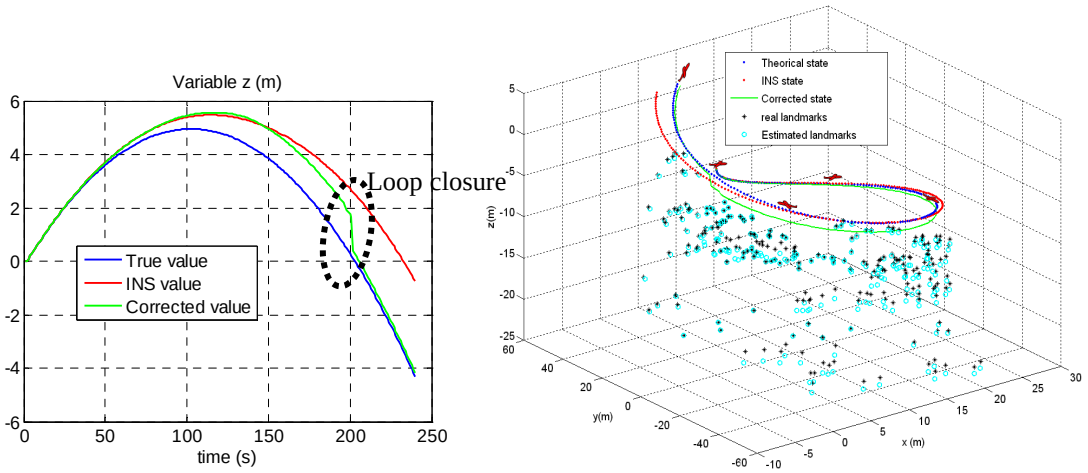
Figure.5. 8 True, INS and corrected UAV trajectory by the EKF SLAM in the case of centred Gaussian noise: a) x estimation, b) y estimation, c) z estimation, d) UAV trajectories with loop closure detection



-a-



-b-



-c- -d-

Figure.5. 9 True, INS and corrected UAV trajectory by the EKF SLAM in the case of centred non Gaussian noise: a) x estimation, b) y estimation, c) z estimation, d) UAV trajectories with loop closure detection

5.9.2 NH∞ Airborne VSLAM

As explained in Section 5.7, the NH∞ estimator does not require specific noise characteristics or a known error covariance matrix. From Figure (5.10. a, b, c, and d) and comparing it to Figure (5.8. a, b, c, and d), we can conclude that if both process and observation noises are assumed centred Gaussian noises then the mean covariance estimator (EKF) could lead to similar (in many cases better) performances than the **minmax** estimator (NH∞). In contrast, the landmarks estimated by the robust NH∞ filter are more accurate than those estimated by the EKF, as can be observed from Table (5.3). Landmark estimation error obtained by the NH∞ filter is very small compared with that obtained by the EKF filter. This could be explained by the linearisation of the highly nonlinear observation model for which the EKF neglects the high order terms. When the assumed characteristics of noises do not hold, which is usually the case for 6DoF navigation and visual data, the NH∞ filter gives much better results than the EKF. The latter conclusion can be validated by comparing Figure (5.9. c, d) with Figure (5.11. c, d). From Figure (5.11.d), it is clear that the estimated trajectory given by the NH∞ is more accurate than that obtained by the EKF, Figure (5.9.d). At ***t=200s*** a loop-closure is detected and at this time a significant improvement of the accuracy is observed especially in Figure (5.11. c) (dashed ellipse). Furthermore, the landmarks

estimated by the robust NH∞ filter, for non centred Gaussian noises, are more accurate than those estimated by the EKF as shown in Table (5.4). As can be seen from this table, the landmark position accuracy given by the NH∞ filter decreases compared to Table (5.3). However, it is still much better than that obtained by the EKF filter, for which the landmark position error raises up above one meter. This is a large error taking into consideration the short navigation time (≈200s).

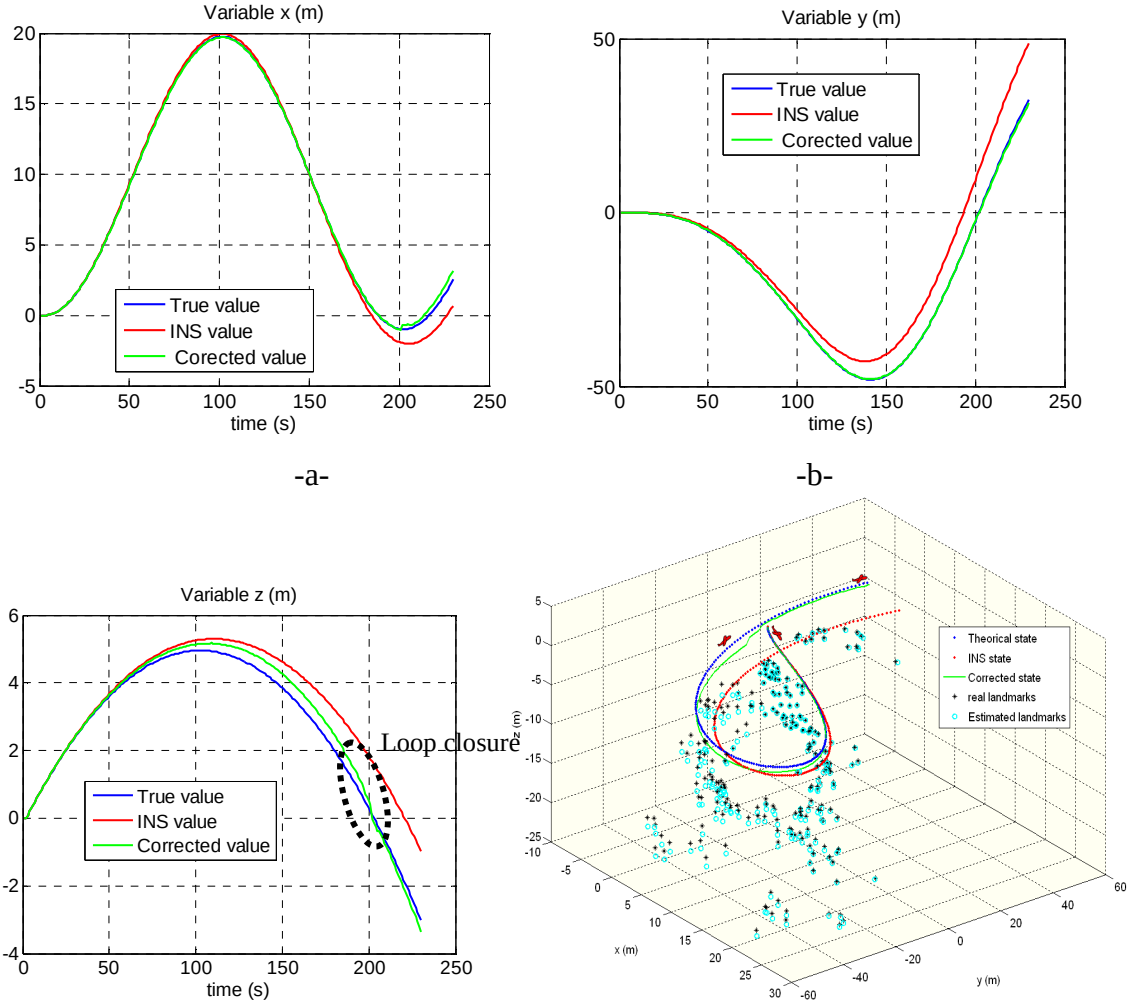


Figure.5. 10 True, INS and corrected UAV trajectory by the NH∞ SLAM in the case of centred Gaussian noise: a) x estimation, b) y estimation, c) z estimation, d) UAV trajectories with loop closure detection

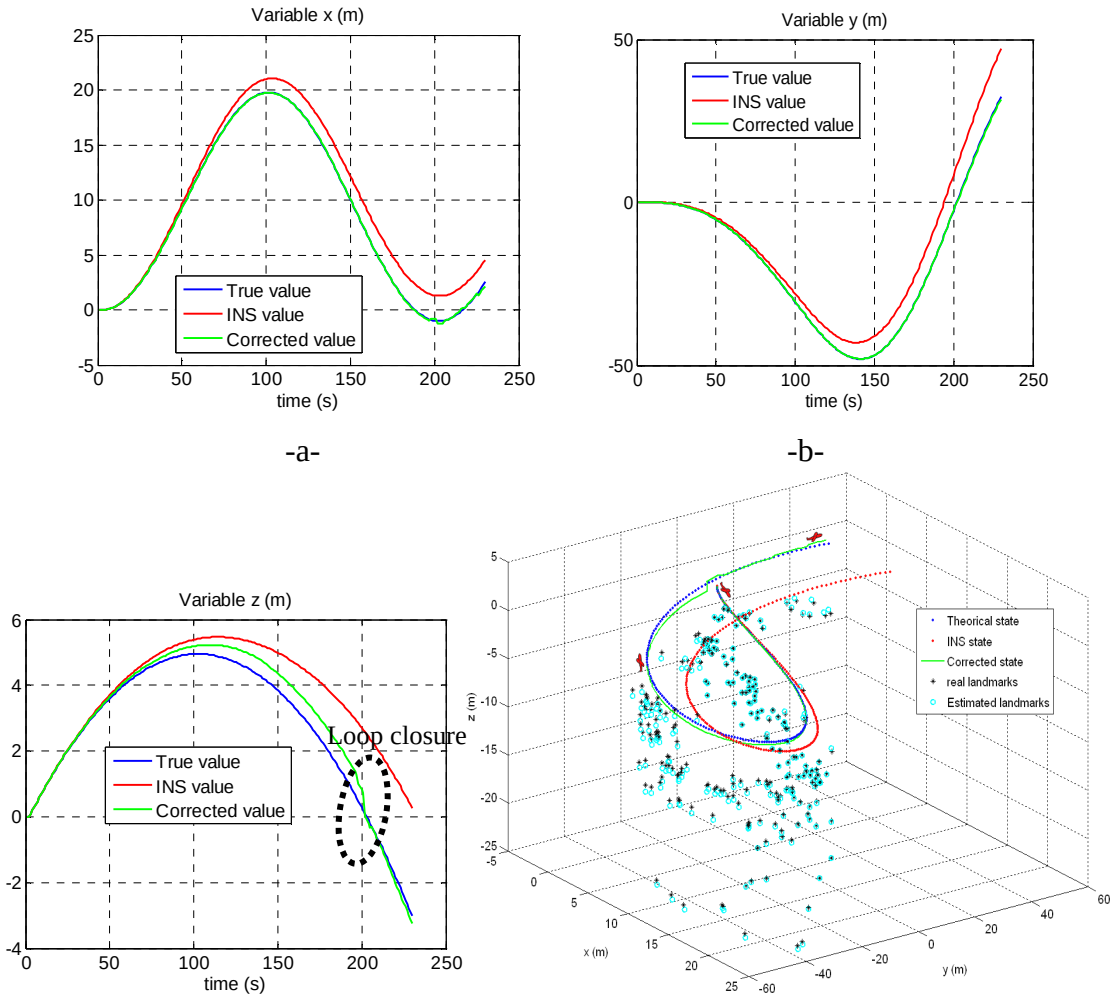


Figure.5. 11 True, INS and corrected UAV trajectory by the NH∞ SLAM in the case of centred non Gaussian noise: a) x estimation, b) y estimation, c) z estimation, d) UAV trajectories with loop closure detection

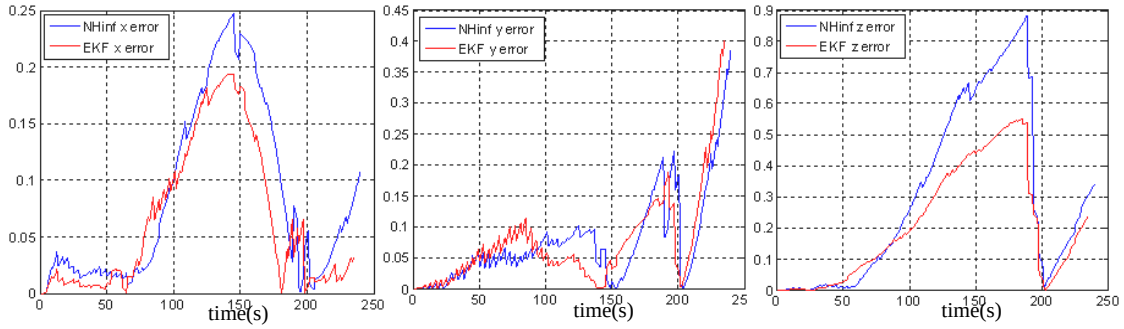
Landmarks error	x-error		y-error		z-error	
	mean	std	mean	std	mean	std
NH∞	0.0047	0.1312	0.0059	0.2221	0.1695	0.3832
EKF	0.1861	0.2578	0.1580	0.2207	0.2929	0.3320

Table 5. 3 comparison between NH∞ and EKF Landmarks accuracy estimation

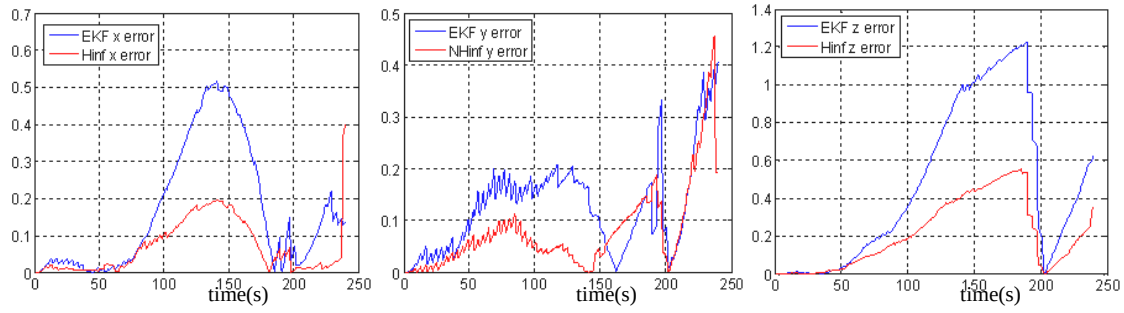
Landmarks error	x-error		y-error		z-error	
	mean	std	mean	std	mean	std
NH∞	0.4692	0.6590	0.3832	0.6850	0.2686	0.5256
EKF	0.5584	1.1404	1.0768	1.3247	0.5998	0.7492

Table 5. 4 comparison between NH∞ and EKF Landmarks accuracy estimation, with non-centred Gaussian noise

In Figure 5.12 below, are illustrated the coordinates of x, y and z representing the UAV position errors in Gaussian and non Gaussian noise cases and using the EKF and NH∞ filters. Confirming the results of the previous section, if the noise is a centred white Gaussian the EKF may perform better than the NH∞ filter, Figure (5.12.a). The advantage of the NH∞ can be observed especially when the noise is non centred or non Gaussian, Figure (5.12.b).



a) Gaussian noise case



b) Non Gaussian noise case

Figure.5. 12 UAV Absolute position estimation error by EKF and NH∞

Figure (5.13) presents the NH∞ and EKF Airborne VSLAM results for a highly nonlinear trajectory. The NH∞ filter shows more robustness and performs, as expected, much better than the EKF filter. This can be explained by fact that the NH∞ filter takes

into consideration all the high order terms of the Taylor expansion (these term should be bounded), while the EKF filter neglects these terms.

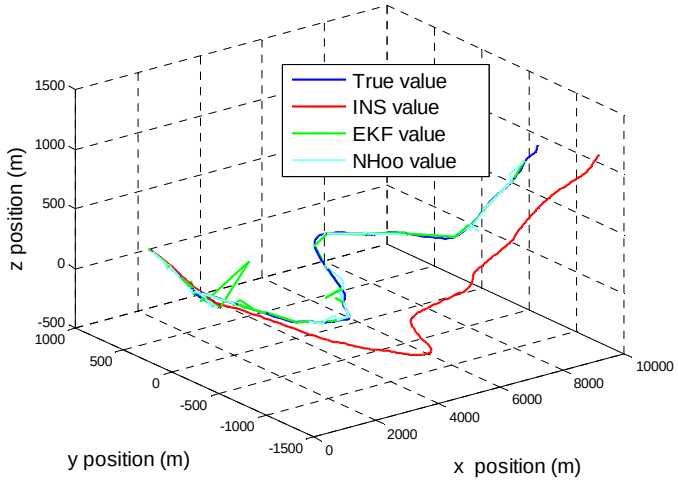


Figure.5. 13 EKF and NH∞ estimator for a highly nonlinear trajectory

5.9.3 Map Management Results

The following figures show the results obtained by the proposed map management approach (***k* -nearest landmarks)**. As can be seen from Figure (5.14), the loop closure is correctly detected at ***t=200s*** despite the fact that the size of the state vector is limited to (***k* + *number of new landmarks***). This implies a significant decrease in the complexity of the algorithm since we choose ***k=20*** with an average of 4 new landmarks observed at each time. Figure (5.15) shows the results obtained by the Stochastic Map (SM) approach. From this figure, we remark that the loop closure is also detected on time but the size of the state vector increases along with the observation of new landmarks. In this case ***232 landmarks*** were observed, which means that the size of state vector is equal to $\mathbf{9_{uav\ states}+232x3_{\ landmark\ states}=705}$ states after only 4 minutes of navigation. This is very heavy and not practical compared to our map management solution where the size of state vector is equal to $\mathbf{9_{uav\ states}+24x3_{\ landmark\ states}=81}$ states only. Furthermore, this state vector size stays small and fixed independently of the navigation time. This improvement can be clearly seen in Figure (5.16) where the size of the state vector is shown increasing linearly with time in the SM approach while it keeps quasi constant in our approach.

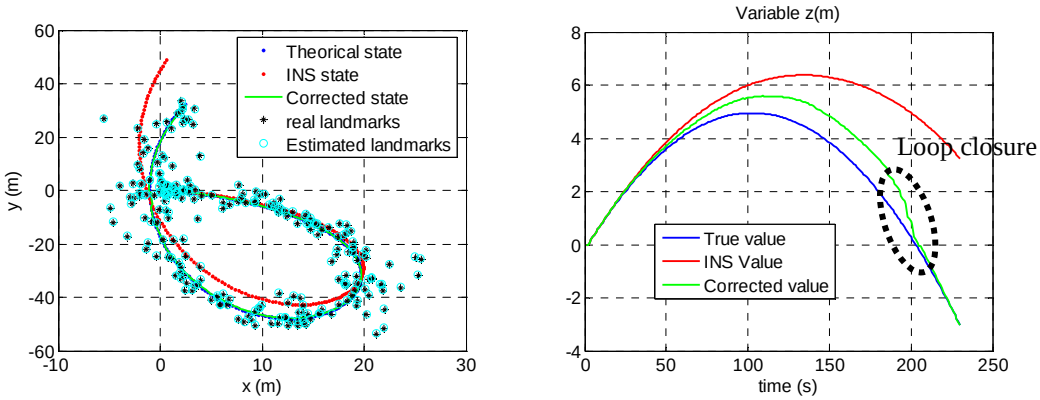


Figure.5. 14 3DVSLAM Map management, approach of the k -nearest landmarks Loop closure detection despite the limited size of the state vector ($x_m \leq 24$ landmarks)

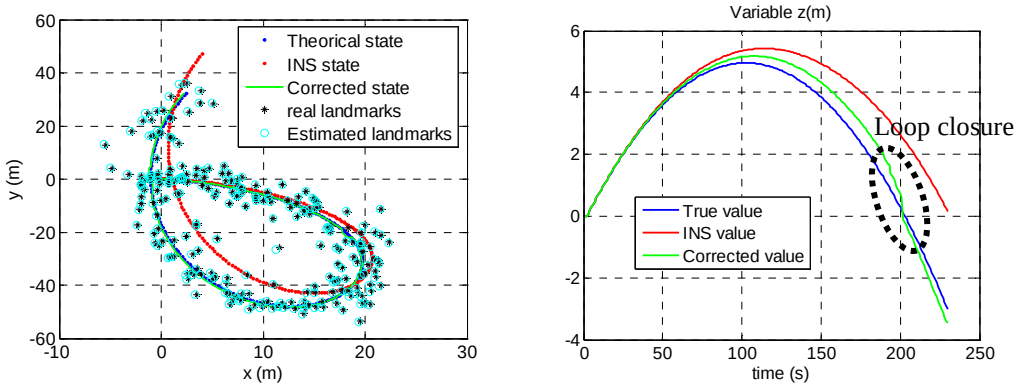


Figure.5. 15 3DVSLAM Map management, augmented map Loop closure detection, (232 landmarks)

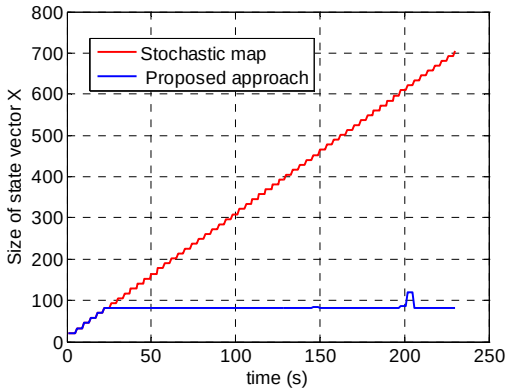


Figure.5. 16 complexity comparison between classical stochastic map and the proposed *k* -nearest landmarks map

5.10 Experimental Results

The experimental setup to validate our robust Airborne VSLAM solution is composed of UAV system with IMU and stereo vision camera as shown in Figure (5.17).

a) Lateral image b) Front image

Figure.5. 17 UAV (COLIBRI) system used for experimental test Comes from Polytechnic School of Madrid

where the IMU characteristics are:

Name	Description	Type
p(Rad/s)	Body frame roll rate in radians. Positive *p* is rolling right wing down. The actual range of the reading should be ±90deg, although very high readings can result from bad gyro bias calibrations.	Double
q(Rad/s)	Body frame pitch rate in radians. Positive *q* is pitching nose up	Double
r(Rad/s)	Body frame yaw rate in radians. Positive *r* is rotating clockwise when viewed from above.	Double
ax(m/s/s)	Forward accelerometer reading in m/s/s.	Double
ay(m/s/s)	Sideways accelerometer reading in m/s/s.	Double
az(m/s/s)	Downwars accelerometer reading in m/s/s.	Double

Table 5. 5 IMU characteristics

and the camera parameters are:

Stereo camera	Intrinsic parameters				Baseline
	α_u	α_v	u_0	v_0	
Camera right	557.6348	557.0185	166.1362	116.9926	20 cm
Camera left	563.5180	563.0117	171.6088	138.1079	

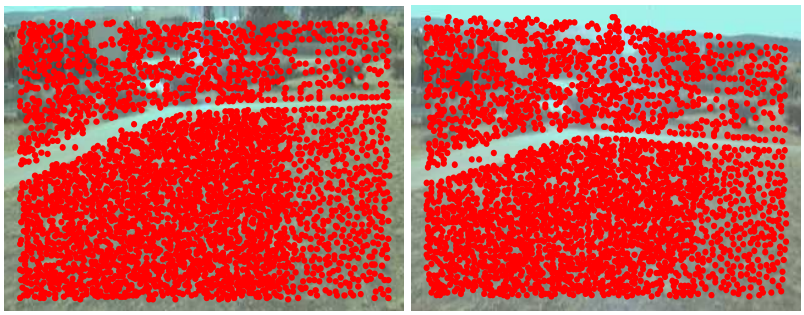
Table 5. 6 Camera parameters

5.10.1 Feature Extraction

Figure (5.18) shows two images acquired by the UAV stereo vision system. Figure (5.19) shows the features extracted using the classical SIFT. As can be seen, too many features are detected which is not suitable for our VSLAM algorithm. Suitable number of features is extracted by the Adaptive SIFT algorithm as shown in Figure (5.20) with energy threshold $E_{th} = 30$.

a) Image left b) Image right

Figure.5. 18 Stereo image



a) Image left b) Image right

Figure.5. 19 SIFT features

a) Image left b) Image right

Figure.5. 20 Adaptive SIFT features with $E_{th} = 30$

5.10.2 Feature Matching

The results of feature matching using different criteria are presented in the following experiments with $DistRatio = 0.4$. From Figure (5.21) and Figure (5.22), very similar matching results are obtained with orientation and 2-norm distances. The number of matches shown achieved is high and may increase the possibility to get false matching. On the other side, the use of $\infty - norm$, in Figure (5.23), allows more distinctiveness of matched features since only a few but robust and distinctive features are detected and matched. This is very suitable for Visual SLAM, which is our intended application.

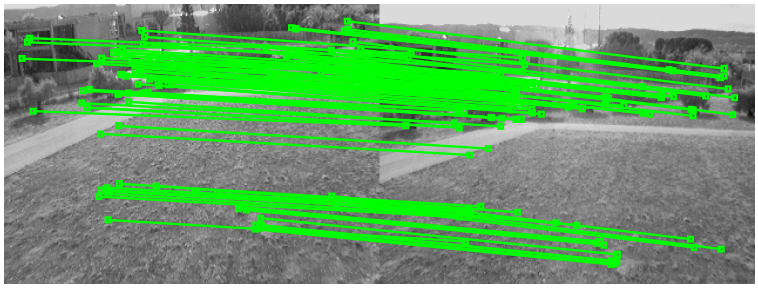


Figure.5. 21 Matching using orientation criteria, 98 matches found

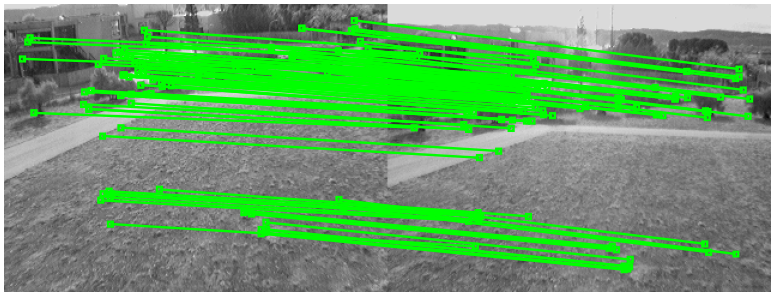


Figure.5. 22 Matching using Euclidean distance, 93 matches found

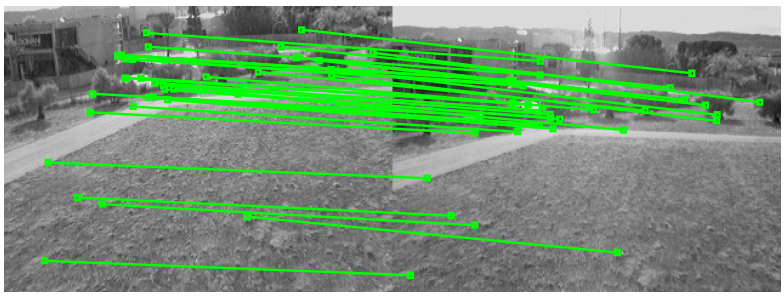


Figure.5. 23 Matching using L-infinity Norm criteria, 36 matches found

The energy of the DoG images in Adaptive SIFT with an energy threshold $E_{th} = 80$ is illustrated in Figure (5.24) for both the left and the right images shown in Figure (5.18). As shown in this figure, when the energy of the DoG image decreases under the threshold E_{th}, a significant increase of scale is applied which implies a rise of the image energy. This process is repeated until the energy of the DoG image exceeds the threshold E_{th} or the scale exceeds a predefined value $\sigma_{\max} = 20$.

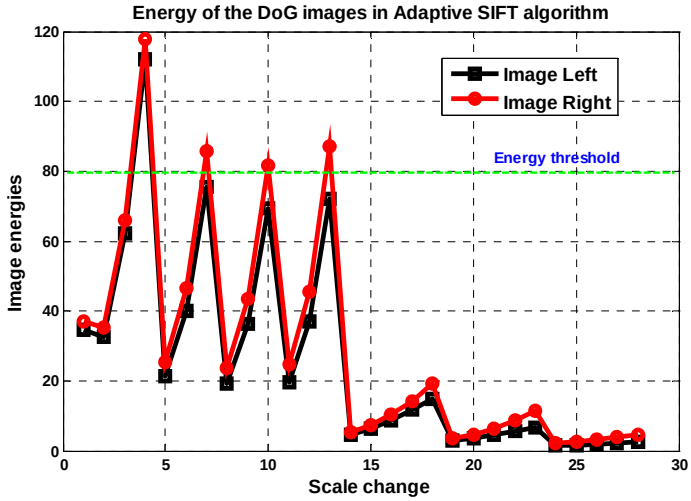
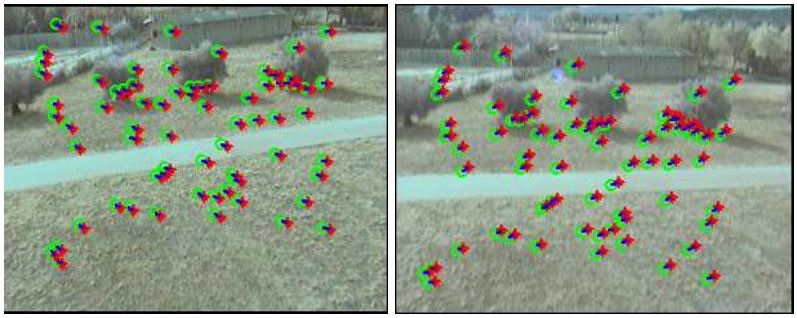


Figure.5. 24 Evolution of the energy of the DoG image in the Adaptive SIFT

Figure (5.25) shows the extracted features in a stereo image at two successive times, red features at *t1* green features at *t2*. As can be seen all the features are well matched which proof the robustness of the adaptive SIFT.



a) Image left b) Image right

Figure.5. 25 Feature association

5.10.3 UAV Localisation and Mapping

Several tests have been made using the helicopter (Colibri) shown in Figure (5.17). In these tests, a series of trajectories around a 3D scene were performed flying in autonomous mode navigation based on way points and desired heading values. The scene is composed of many objects, including a grandstand, a van and many other elements, and a series of marks suitable for feature and corner detection. For each flight test a 30 f.p.s. image sequence of the scene was obtained and associated with the UAV attitude information. This includes the GPS position, IMU data (Heading, body frame angles and displacement velocities) and the helicopter position given by the filtered output on the local plane with reference to the takeoff point.

SLAM principle

1- At *t=0*, UAV position is known (0,0,0,0,0,0).
2- At *t=t1*, the UAV displaces and observes some features.
3- The UAV new pose is estimated using the INS and the new features are initialized in the map using Inverse Observation Model.
4- At *t=t2*, the UAV displace again and the observed features are divided on two sets: new features will be initialised in the map, re-observed features (already observed) will be used to correct the UAV pose given by the INS.
5- Step 4 is repeated, and the loop closing (if exist) will be detected automatically.

5.10.3.1 Airborne 3D EKF VSLAM

UAV position, velocity and orientation estimated by the EKF VSLAM algorithm are presented in Figure (5.26). As can be seen from Figure (5.26. a, b and c), the position estimated by the EKF VSLAM is more accurate than the INS position which drifts significantly with time. UAV velocities in the body frame are given by Figure (5.26. d, e and f). From these figures, we observe that for a smooth trajectory of the UAV, the velocity estimation using INS is quite similar to VSLAM estimator. Figure (5.26.g, h and i) show the UAV attitude, we can observe a significant drift in the INS attitude while the EKF VSLAM maintains acceptable accuracy.

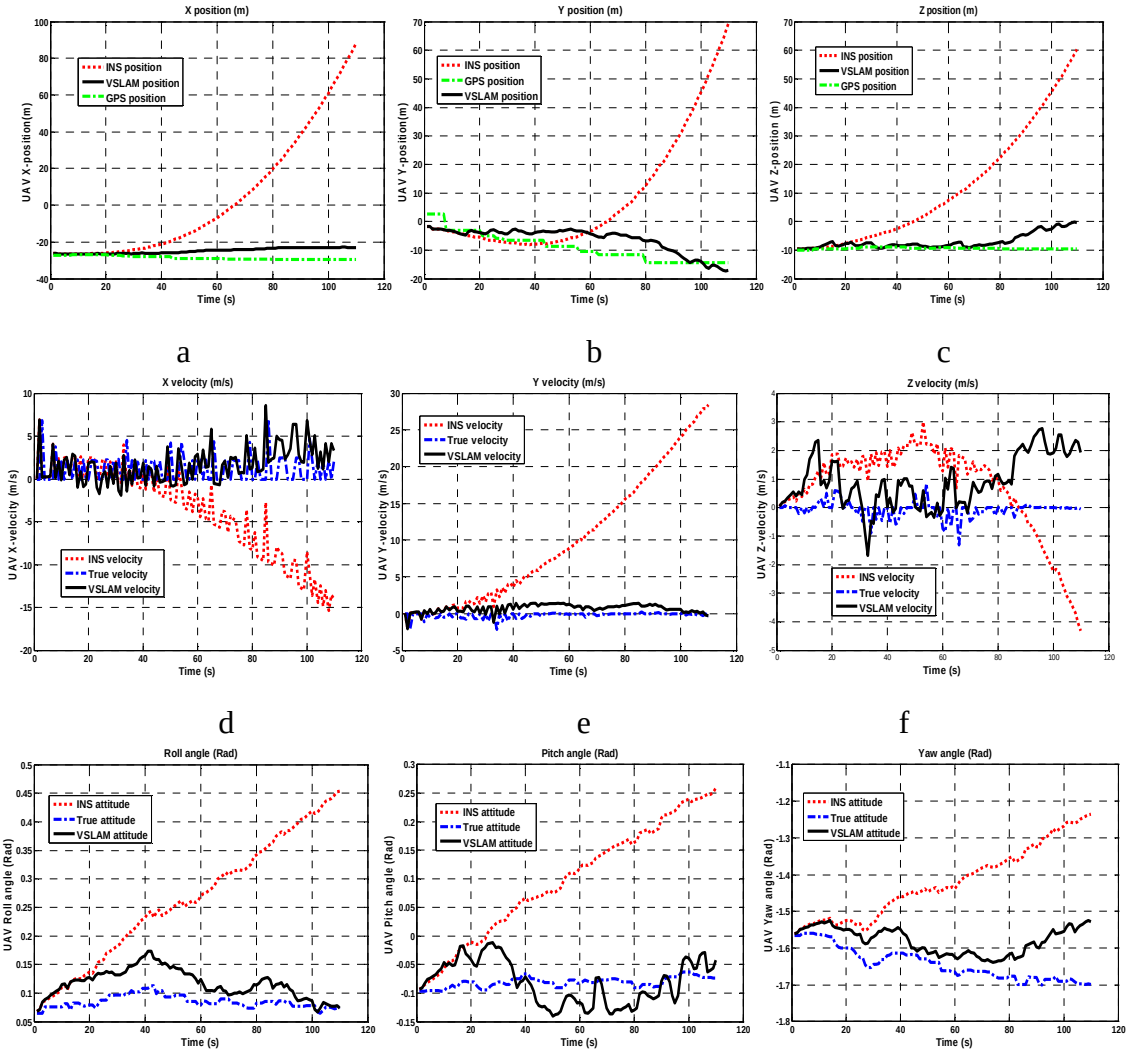


Figure.5. 26 UAV localisation using EKF VSLAM

5.10.3.2 Airborne 3D NH∞ VSLAM

UAV position, velocity and orientation estimated by the NH∞ VSLAM algorithm are presented in Figure (5.27). As can be seen in Figure (5.27. a, b and c) the position estimated by the NH∞ VSLAM is more accurate than the INS position which drift significantly during time. UAV velocities in the body frame are given by Figure (5.27. d, e and f), from these figures we observe that for smooth trajectory of the UAV, the velocity estimation using INS is quite similar VSLAM estimator. Figure (5.27.g, h and i) show the UAV (Roll, Pitch and Yaw) angles, we can observe a significant drift in the INS attitude while the NH∞ VSLAM maintains acceptable results.

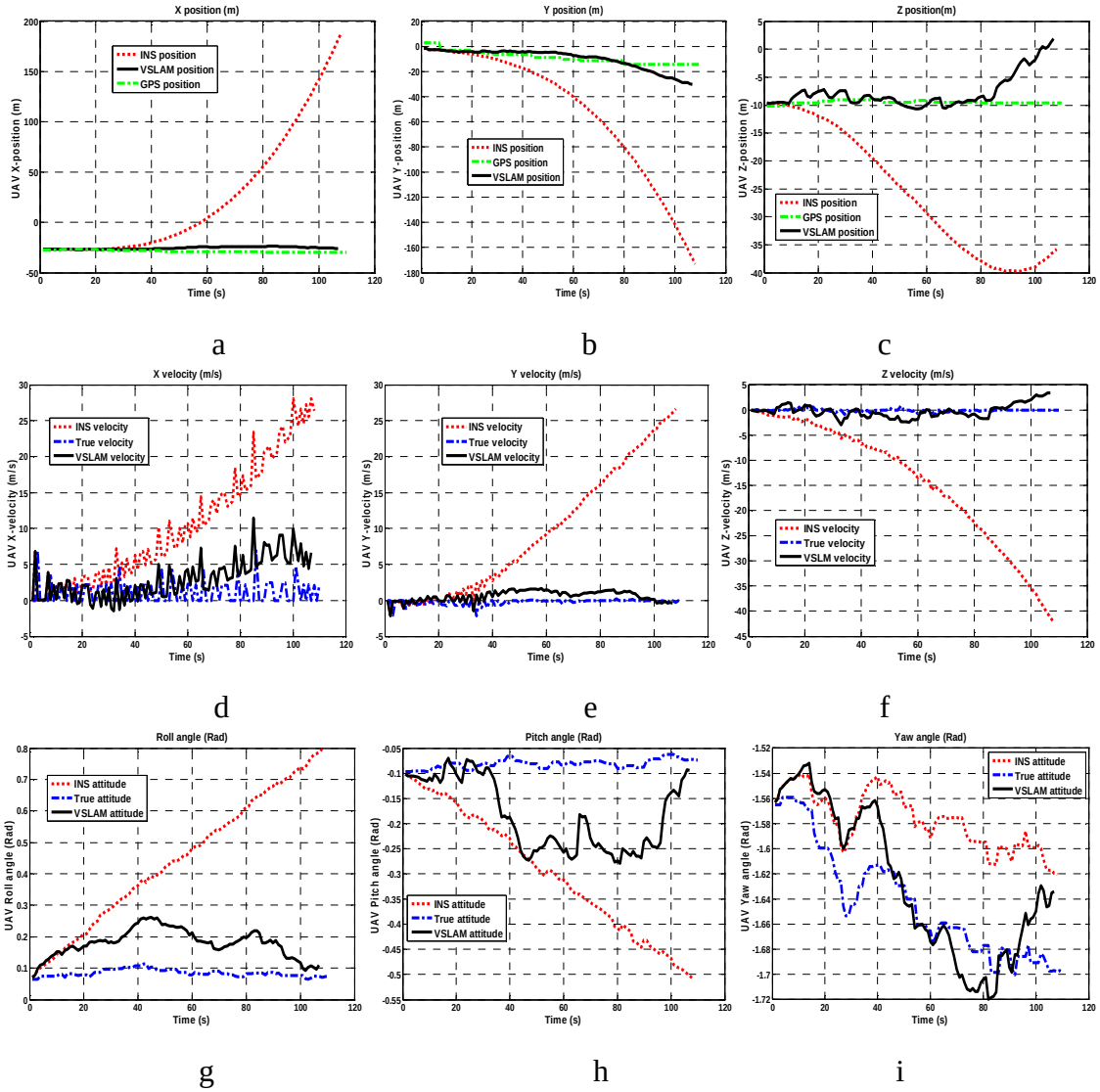
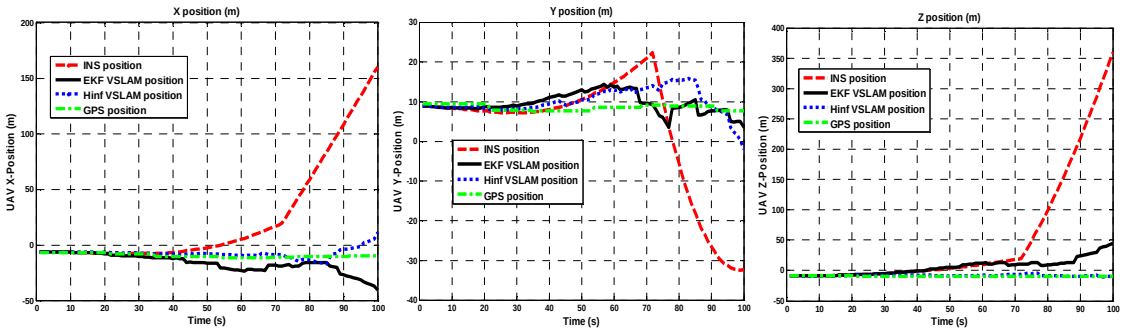


Figure.5. 27 UAV localisation using NH∞ VSLAM

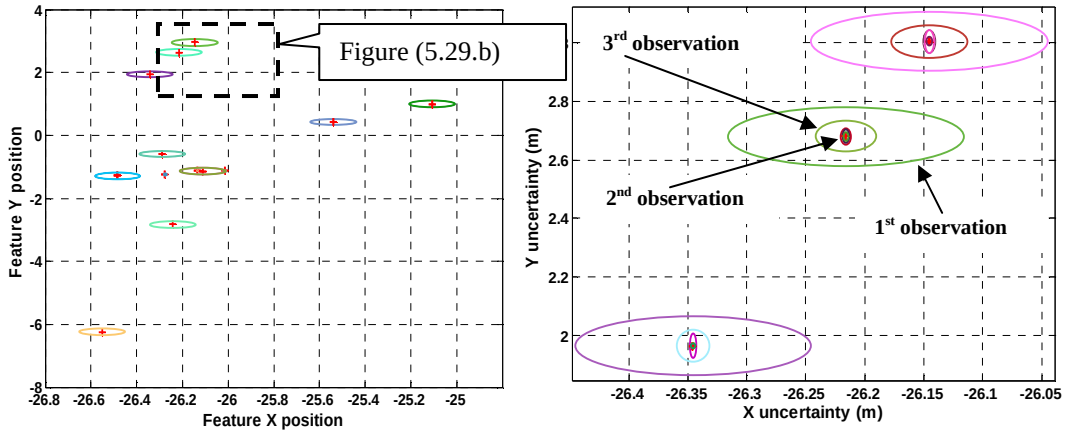
5.10.3.3 EKF VSLAM vs NH∞ VSLAM

Figure (5.28), shows a comparison between the EKF VSLAM and NH∞ VSLAM for the UAV localisation. As explained in Section 5.9, the EKF filter may perform better than the NH∞ when the process and observation noise are white Gaussian and centered. However, in real sensing systems (IMU, and Cameras) the previous condition can rarely hold. In this case, the use of the NH∞, which does not make any assumption on the noise statistics, shows much better navigation and mapping performances than the EKF and provides an accurate position of the UAV as illustrated in Figure (5.28).



a b c

Figure.5. 28 EKF vs NH∞ UAV localisation



a b

Figure.5. 29 Features uncertainties

Extracted feature in the navigation frame are represented in Figure (5.29.a). Feature uncertainties following X (north) and Y (east) are represented in Figure (5.29.b) which is a zoom of the dashed square area of Figure (5.29.a). As can be seen the uncertainty of features decreases when the features are re-observed.

5.11 Conclusion

In this chapter, we proposed a new approach to solve the Airborne VSLAM problem. It is based on the development of a full observation model of an IMU/Stereo cameras embedded on a UAV. Contributions in this chapter range from the observability and consistency analyses of the EKF (*NH*∞) Airborne VSLAM to the proposition of a robust version of an Airborne VSLAM using a new map management approach reducing VSLAM algorithm complexity. *NH*∞ VSLAM is compared favourably with the EKF VSLAM and accurate UAV positions are estimated when no assumptions are made either on the process model or on the noise characteristics.

Chapter 6

Cooperative Visual SLAM

CHAPTER VI

Cooperative Multiple UAV Visual SLAM

6.1 Overview

In this chapter, Cooperative Airborne VSLAM problem is investigated. First, a centralised cooperative VSLAM algorithm with loop closing detection is proposed. Then, a simulation and experimental results are presented with and without map management options. After that, Cooperative VSLAM decentralised architecture suitable for embedded system is proposed and validated using simulation data. Finally conclusions and future works are suggested.

6.2 Introduction

In many applications as the ones related to defence industry, a single sensing platform may not be sufficient to precisely collect data or to create maps of an unknown or partially known environment. Currently and probably more in the future, distributed sensing systems are required to gather precise information about remotely monitoring environments. Fleets of Autonomous Underwater Vehicles (AUVs) [112] and Unmanned Aerial Vehicles (UAVs) [113] have been lately proposed for applications ranging from environmental monitoring to surveillance and defence. These systems require the ability to both share and then fuse information from different sources into a consistent scene view [114]. In fact deploying these multiple vehicles into an environment by providing them with a mechanism for sharing information can deliver higher data rates, increase in robustness, and system failure minimisation.

Autonomous navigation of multiple vehicles has introduced the problem of Cooperative Simultaneous Localisation and Mapping (C-SLAM). C-SLAM is performed when multiple vehicles share navigation and perception sensing information in order to improve their own position estimates beyond what is possible with a single vehicle. Simple collective navigation has been demonstrated in simulation using multiple ‘cartographer’ vehicles that randomly explore their respective environments

[115]. Sty in [116] performs simple relative localisation between collaborators using directional beacons. Other research work presented challenges in terms of localisation performances with growing uncertainties for C-SLAM [117-118]. Simulation and experimental validations were conducted in this later work to support their analysis. In addition, techniques based on entropy minimisation [119], and information theory [120], were also developed for Cooperative SLAM problem.

Up to date a little effort only has been done in the field of multi-vehicle visual SLAM, named as Cooperative Visual SLAM (C-VSLAM). The latter considers the case where several vehicles move within their environment and build their map cooperatively using visual sensors [121-122]. This challenging visual absolute localisation/mapping problem is different from the recent research work aiming to achieve relative navigation in multi-vehicle systems [123]. In this chapter, we concentrate on this C-VSLAM problem and propose a solution that allows us to build a visual map using a set of reliable and a priori unknown visual feature observations obtained by a team of unmanned vehicles.

Some solutions for Cooperative SLAM [124], and more specifically Cooperative Visual SLAM [125-126], are based on Extended Kalman filter. However, this latter is very sensitive to outliers. Moreover, the lower bound for the map accuracy, as presented in [127], is violated due to errors introduced during EKF linearisation process producing inconsistent estimates [127-128]. Achieving Cooperative SLAM, [129-130], and more specifically Cooperative Visual SLAM, [122, 126], based on particle filter estimation scheme present a major drawback related to the computation time that makes it not very suitable for hard real time applications as in airborne navigation. Although some very recent progress has been made by proposing filters that try to approximate the nonlinear SLAM problem [131-132], and lately C-SLAM, [133], by means of what is called Square Root Information Smoothing technique. Issues related to efficient retrieval of marginal covariance and results obtained through few data sets show that there is still a way to go to bring in such techniques to meet real time aerospace visual navigation requirements.

In this chapter, based on a robust Nonlinear H∞ (*NH∞)* sensor fusion algorithm, we propose to solve the Cooperative Visual Simultaneous Localisation and Mapping

problem for Unmanned Vehicles navigating in unknown natural environment, under realistic conditions and with experimental validations. Effective stereo observation model and a map management approaches are also used to ensure a reduced computation time as to maintain a suitable correlation between feature observations, which is very important in C-VSLAM loop closing detection.

6.3 Centralised Cooperative VSLAM (CC-VSLAM)

Centralised C-VSLAM is an extension of the single VSLAM solution proposed in Chapter 5. In multi-vehicle VSLAM problem, the estimated state vector becomes the position, velocity and attitude of the multiple vehicles and the positions of point feature observations in the environment.

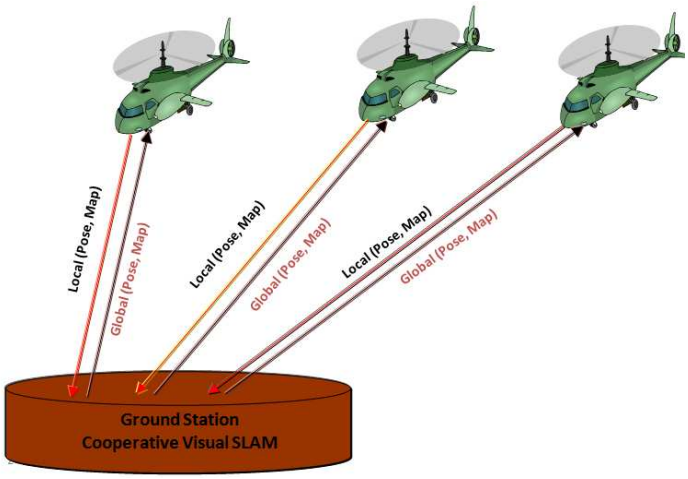


Figure 6. 1 Centralised Architecture

The non-linear discrete time state transition equation given by Equation (5.21) can be extended to the multiple UAVs case as follows:

$$X_k = f(X_{k-1}, U_{k-1}) + g(X_{k-1}) w_{k-1}$$
$$Y_k = h(X_k, v_k)$$

f is the discrete time state transition function, X_k is the state vector at time step k, w_k is some additive process noises, Y_k is the observation made at time k by all the UAVs, v_k is some additive observation noises. The objective of the filtering technique is, then, to estimate X_k using available observation Y_k. The CC-VSLAM state vector and Jacobian matrix will be given by:

- State vector,

$$X_k = \begin{bmatrix} X_{uav1,k} \\ \vdots \\ X_{uavM,k} \\ m^n_{1,k} \\ m^n_{2,k} \\ \vdots \\ m^n_{N,k} \end{bmatrix}$$

- Jacobian matrix of f **w.r.t** X_k

$$F_{sys} = \frac{\partial f}{\partial X_k} = \begin{bmatrix} F_{uav1,k} & 0_{9\times9} & \cdots & 0_{9\times3N} \\ 0_{9\times9} & \ddots & 0_{9\times9} & \vdots \\ \vdots & 0_{9\times9} & F_{uavM,k} & 0_{9\times3N} \\ 0_{3N\times9} & \cdots & 0_{3N\times9} & I_{3N\times3N} \end{bmatrix}$$

- Jacobian matrix of f **w.r.t** w_k

$$\Gamma_{sys} = \frac{\partial f}{\partial w_k} = \begin{bmatrix} \Gamma_{uav1} & 0_{9\times6} & 0_{9\times6} \\ 0_{9\times6} & \ddots & 0_{9\times6} \\ 0_{9\times6} & 0_{9\times6} & \Gamma_{uavM} \\ 0_{3N\times6} & 0_{3N\times6} & 0_{3N\times6} \end{bmatrix}$$

- Jacobian matrix of h **w.r.t** X_k

$$H_{sys} = \frac{\partial h}{\partial X_k} = [0_{4\times9(i-1)} \quad H_{uav\_i} \quad 0_{4\times9(M-i)} \quad 0_{4\times9(j-1)} \quad H_{feature\_j} \quad 0_{4\times9(N-j)}]$$

For the i^{th} UAV and the j^{th} feature.

- Jacobian matrix of f **w.r.t** v_k

$$V_{sys} = [I_{4\times4}]$$

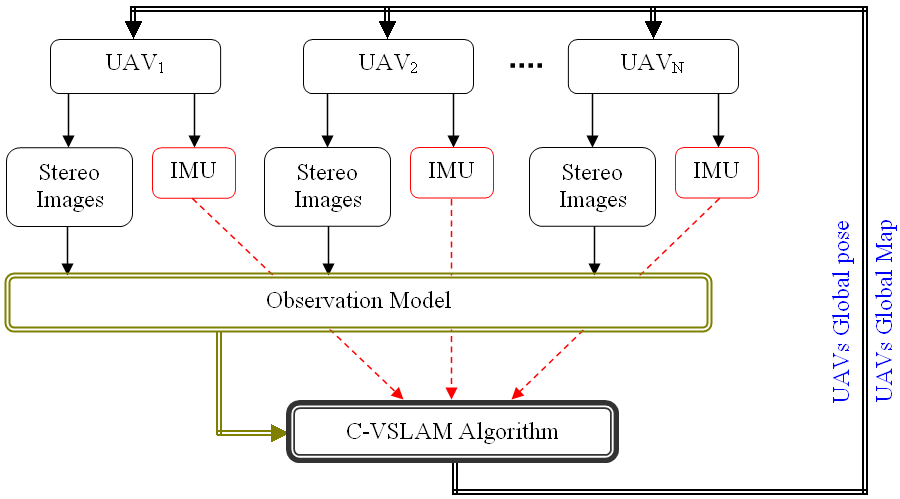


Figure 6. 2 Cooperative VSLAM architecture

Figure (6.2) shows a diagram of our cooperative VSLAM concept. The essential step in this architecture is the Observation Model as a good cooperation is obtained when the shared region between the unmanned aerial vehicles (UAVs) is large. The shared features will used to update the global map and then improve the global pose of UAVs.

6.3.1 Airborne Cooperative Visual SLAM

To detail our Cooperative VSLAM concept, assume we have ***N*** UAVs navigating in outdoor environment, and ***Mi*** is the number of features observed by the i^{th} UAV at time $t=k$. The C-VSLAM algorithm runs centrally at the ground station and communicates the position and the map to each UAV as follows:

At $t=0$ the UAVs positions, velocities and orientations are assumed known as well as the covariance matrix. During navigation, each UAV observes a set of features that can be divided into three types Figure (6.3):

Type 1: feature re-observed (has been observed by the same UAV_i).

Type 2: feature re-observed (has been observed by other UAV_j and $j \neq i$).

Type 3: new feature observed for the first time.

Features of *Type 1* and *Type 2* will be used to update the map and the UAVs states, where features of *Type 3* will be initialised and added to the map using the inverse model of observation [100]. When a new feature is observed by more than one UAV then it will be initialised more accurately (red feature in Figure (6.3)).

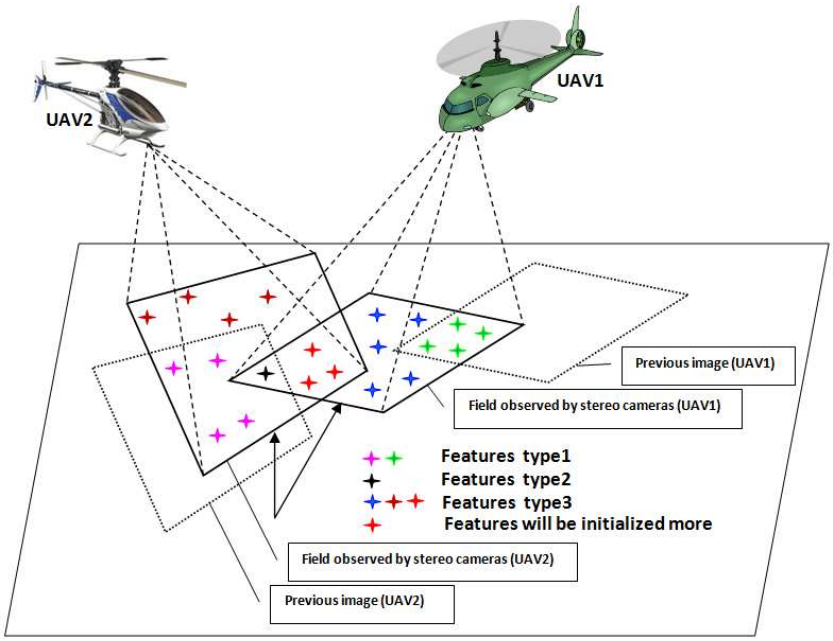


Figure 6. 3 Features in C-VSLAM algorithm

6.3.2 C-VSLAM algorithm

Cooperative VSLAM algorithm is presented below. It is similar to the Single VSLAM with some differences in feature observations and state updates.

- ***Initialisation:***

$$U_0 = \left[U_{uav1}(0),...,U_{uavN}(0)\right]^T$$

$$X_0 = [X_{uav1}(0),..., X_{uavN}(0)]^T$$

$$P_0 = diag([P_{uav1}(0),...., P_{uavN}(0)])$$

$$Q = diag([Q_{uav1},..., Q_{uavN}])$$

$$R = I_{4\times 4}$$

- ***Prediction***

$\hat{X}_{k+1/k} = f(\hat{X}_{k/k}, U_k)$ ***%State prediction using process model***

$P_{k+1/k} = F_k P_{k/k} F_k{}^T + \Gamma_k Q_k \Gamma^T{}_k$ ***% Covariance prediction***

- ***Update***

$\hat{y}_k = H_k X_{k/k-1}$ ***% predicted measure***

$$y_k = [\cdots \quad y_k^{i,j} \quad \cdots] = \begin{bmatrix} \cdots & u_{left}^{i,j} & \cdots \\ \cdots & v_{left}^{i,j} & \cdots \\ \cdots & u_{right}^{i,j} & \cdots \\ \cdots & v_{right}^{i,j} & \cdots \end{bmatrix}$$ ***%coordinates of the j^{th} feature observed by the i^{th} UAV***

- ***For i=1 to N*** ***%number of UAVs***
  - ***For j=1 to Mi*** ***%number of feature observed by UAVi***
  - ***Find correspondence between*** $y_k^{i,j}$ ***and*** $\hat{y}_k$
  - ***If correspondence is found***
    - $K_k = P_{k/k-1} H_k (H_k P_{k/k-1} H^T{}_k + R_k)^{-1}$
    - $\hat{X}_{k/k} = \hat{X}_{k/k-1} + K_k (y_k^{i,j} - \hat{y}_k^l)$
    - $$\Lambda_{k/k} = \begin{bmatrix} P_{k/k-1} - \gamma^2 I & -P_{k/k-1} H_k^T \\ -H_k P_{k/k-1} & H_k P_{k/k-1} H_k^T + R \end{bmatrix}$$
    - $$P_{k/k} = P_{k/k-1} - P_{k/k-1} \begin{bmatrix} -I & H_k^T \end{bmatrix} \Lambda_{K/K}^{-1} \begin{bmatrix} -I \\ H_k \end{bmatrix} P_{k/k-1}$$
  - ***Else*** $y_k^{i,j}$ ***is a new feature to be added to the map using the inverse observation model***
  - ***End if***
  - ***End for***
- ***End for***

- ***Initialize the new features in the map***

C-VSLAM algorithm

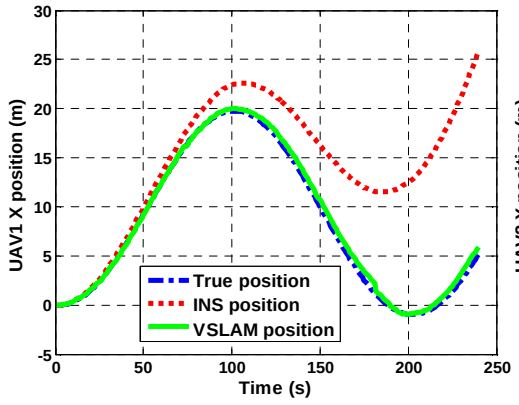
6.3.3 Loop Closing Concept in Multiple VSLAM Case

One key requirement for VSLAM to work well is feature re-observation. This has two advantages: the improvement of the feature location estimates in the map and the improvement of the vehicle location estimates. The latter is due to the statistical correlations that are built up between the environment features and the vehicle.

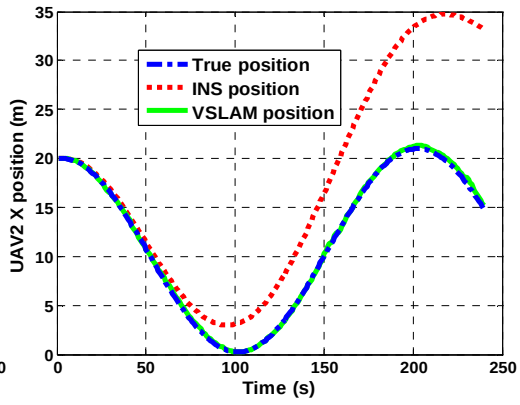
Based on re-observation of features, long term VSLAM navigation is possible as loop closing detection (re-visiting the same area) can be used. This is also necessary to improve the consistency of the NH∞ VSLAM algorithm as shown in [100]. The consistency of the NH∞ VSLAM is improved when a loop closing is detected. In multiple UAV case two loop closing concepts can be defined. Loop closing when the UAV observes already observed features and Loop closing effect due to UAV cooperation. The former is the same concept as classically presented for single VSLAM. The latter happens when one cooperating UAV observes features already observed by one or more cooperating UAVs. Furthermore, when a new feature is observed by more than one UAV, then the initialisation of that feature in the global map is more accurate.

6.4 Simulation results

This section presents simulation results of C-VSLAM involving two UAVs. Each UAV has its own inertial measurement unit (IMU) and stereo vision cameras. The C-VSLAM algorithm is simulated to run centrally at the command station while communicating the position and the map to each UAV.



-a-



-d-

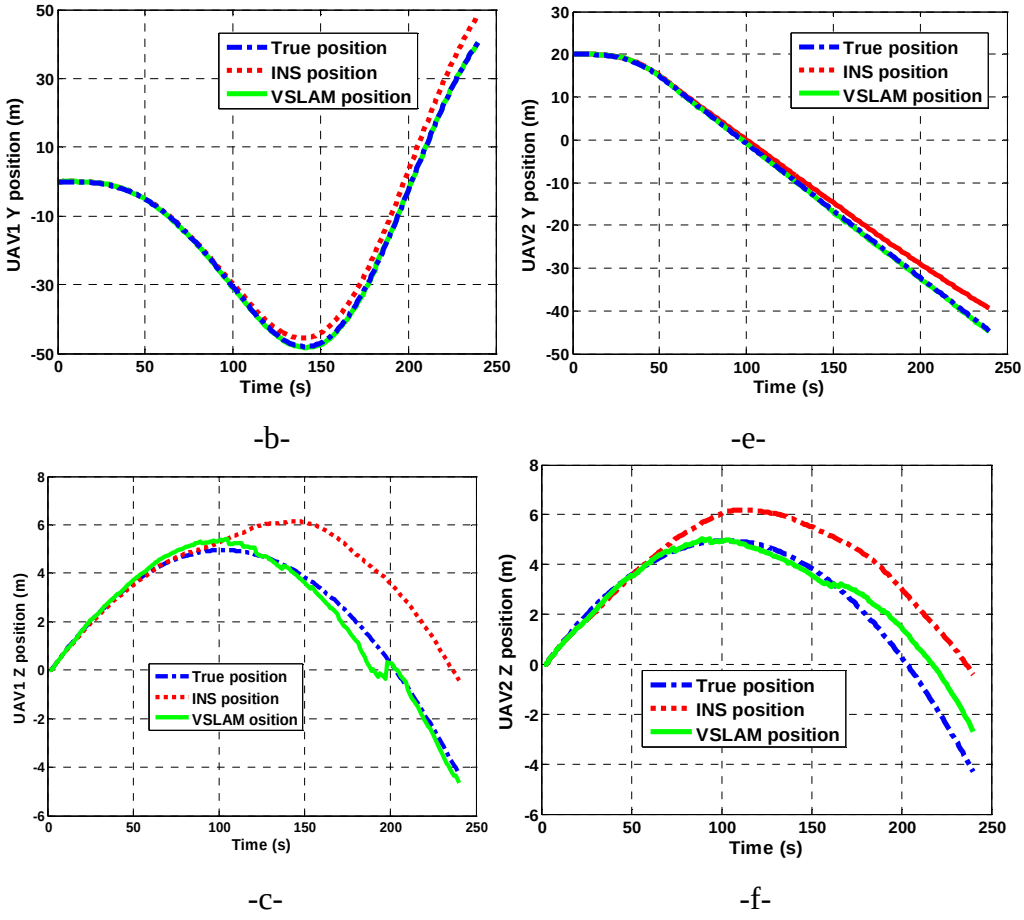


Figure 6. 4 UAVs positions, left XYZ position of UAV1 in navigation frame Right XYZ position of UAV2 in navigation frame

In Figure (6.4), curves in the left (right) side show the position of the UAV1 (UAV2) in the X, Y and Z axes. As can be seen, X position Figure (6.4. a, d) and Y position Figure (6.4. b, e) are estimated with significant accuracy. This can be explained by the fact that cameras or stereo vision system can provide precise bearing information. This is not completely the case for the range information where the stereo vision system provides less accurate Z position as shown in Figure (6.4.c, f). On the above figures, we can also observe the effect of loop closing detection on UAV1 at ***t=200s***, as well as the precision improvement obtained when UAV2 visits features already visited by the UAV1 at ***t1=80s*** and at ***t2=150s.***

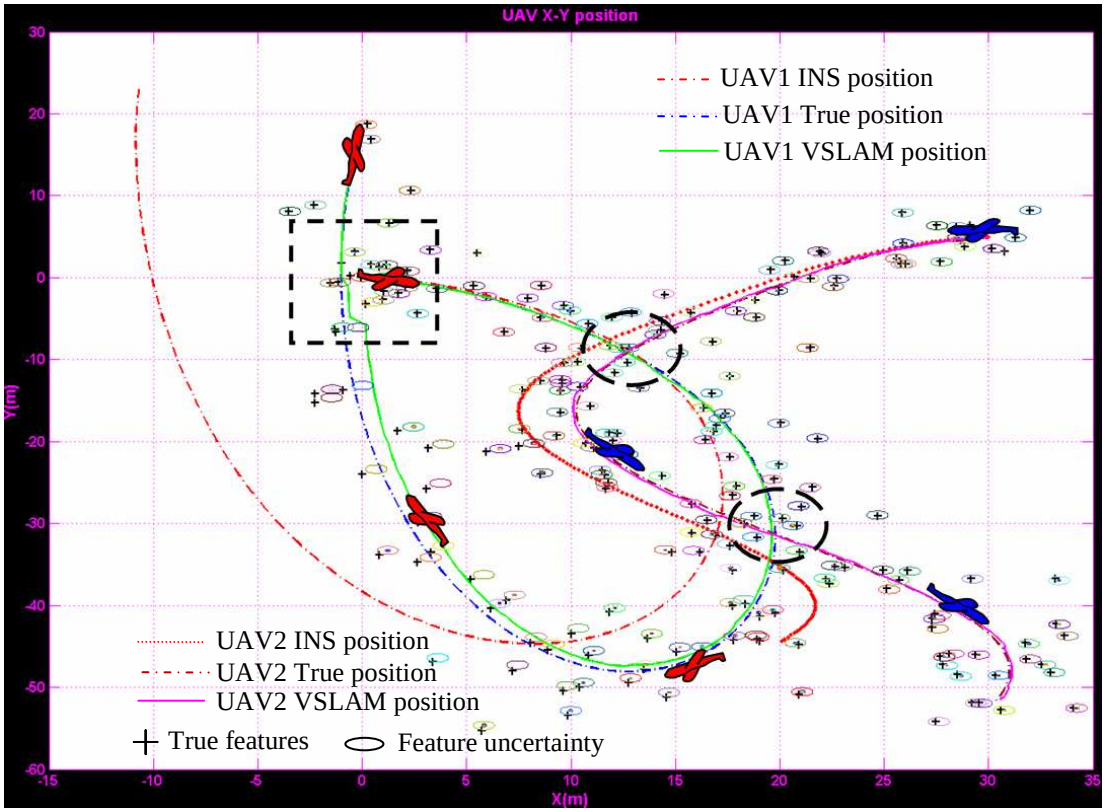


Figure 6. 5 UAV1 and UAV2 True, INS and corrected position

Figure (6.5) shows the trajectories of the two UAVs in the X and Y axes. While UAV1 (Red) closes its loop at ***t=200s***, (dashed square), UAV2 (Blue) does not make any loop closing but it visits many features already visited by UAV1 (dashed ellipses).

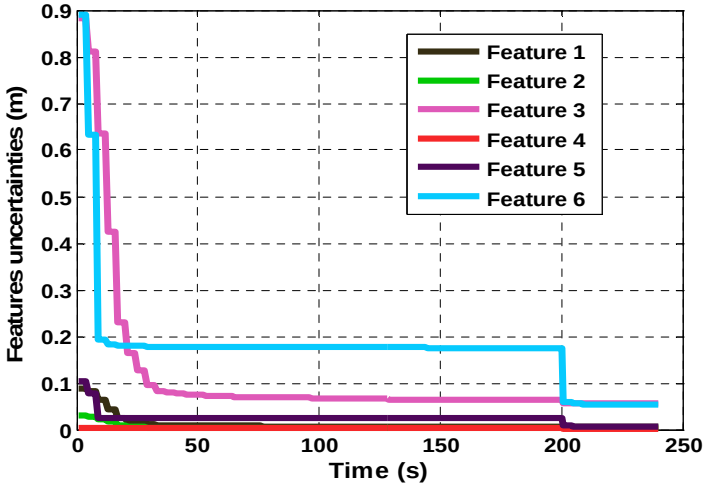


Figure 6. 6 Features uncertainties

Figure (6.6) presents the evolution of the uncertainties for six features from the global map. As shown, the uncertainty of each feature decreases with time. At ***t=200s*** a significant decrease of the uncertainty is observed and this is justified by the loop closing detection, which improves the consistency of the estimator at that time.

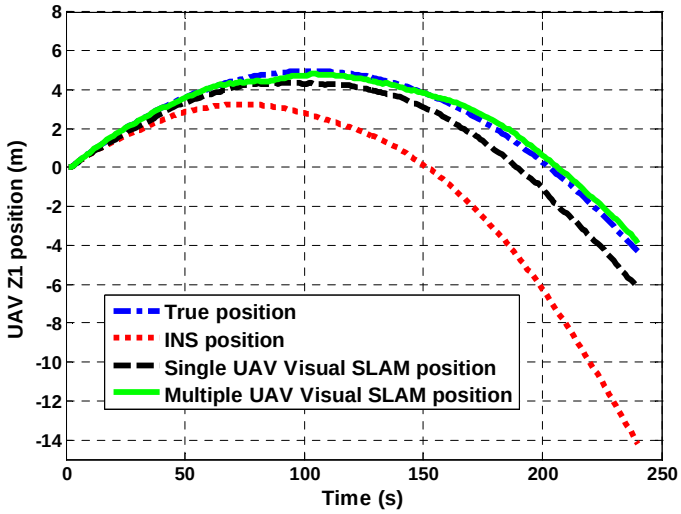


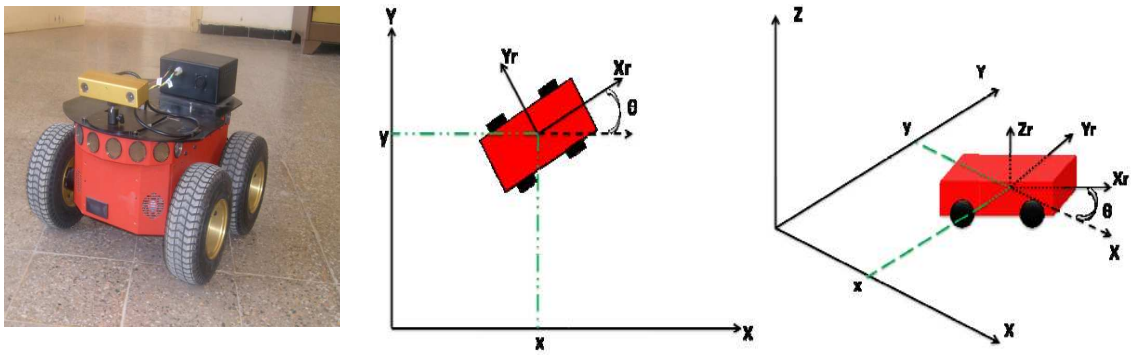
Figure 6. 7 UAV Z estimation single UAV SLAM vs multiple UAV SLAM

Figure (6.7) shows a comparison between single and cooperative UAV VSLAM in simulation. The estimation of the UAV1 Z-position with a single VSLAM, even if it still much better than the INS position, leads to an increasing error with time if no loop closure is detected. On the other hand, the estimation of the UAV1 Z-position with cooperative VSLAM provides more accurate position.

6.5 Experimentations

6.5.1 Experimental setup

Since multiple UAV setup was not available, experimental validation was arranged based on cooperative mobile robots (Pioneer3 AT), Figure (6.8.a), without losing generality in the validation process from ground robots (3DOF) to aerial vehicles (6DOF). In this experiment, stereovision data are fused with Odometer pose instead of the UAV full INS information.

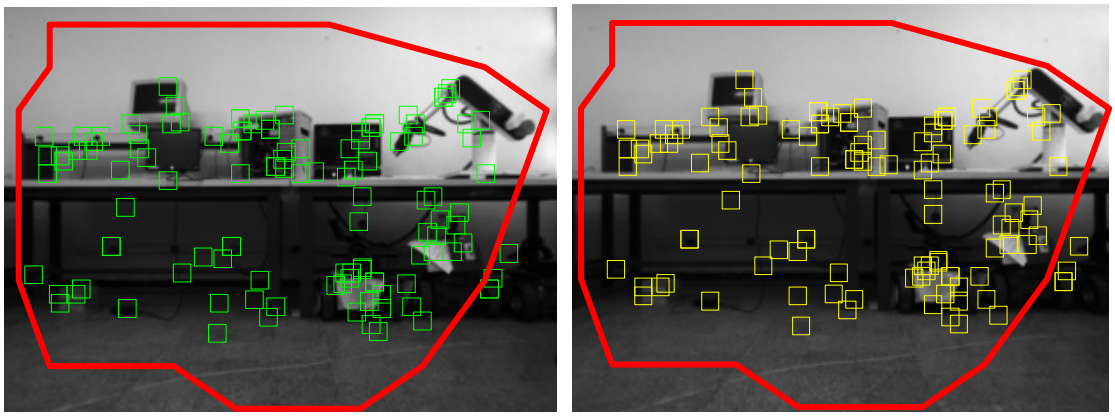


-a- **-b-** **-c-**

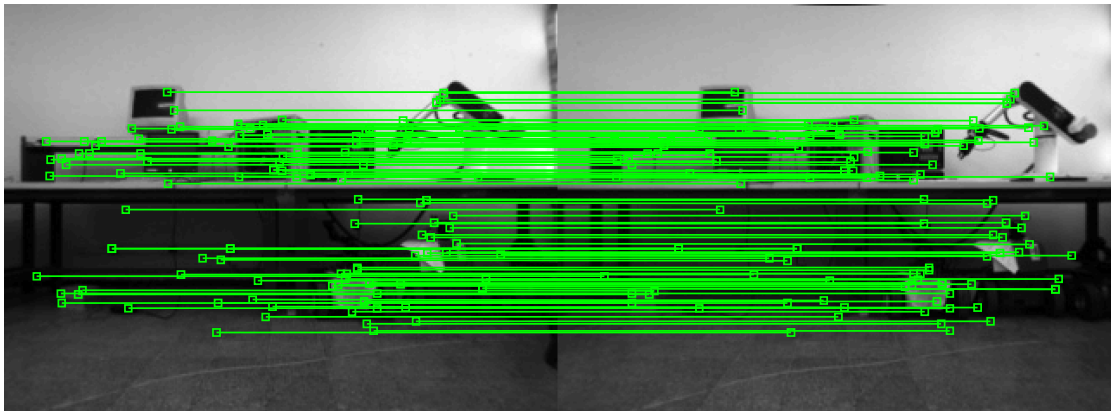
Figure 6. 8 Mobile robot representation in navigation frame

6.5.2 Feature Extraction and Matching

Feature extraction and matching for robot 1 and 2 based on the development done in Section (4.2.8.1) are presented in Figure (6.9). One essential thing for robot cooperation is the availability of shared features (limited region by the red line) as in Figure (6.9.a, b, d and e). The shared features between robots are very important to maintain suitable accuracy and consistency even without closing the loop. The observation of features already observed by others robots (not necessary the same robot) provides similar rewards provided by the loop closing detection.



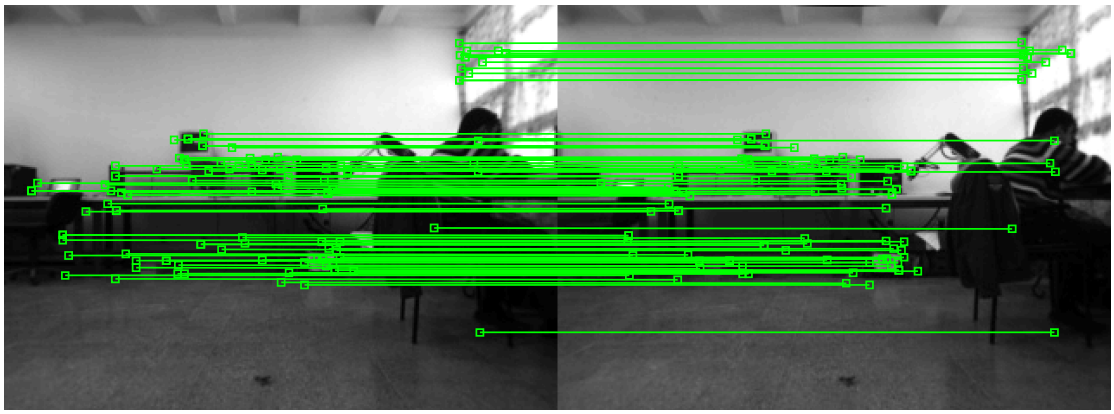
a b



c

d

e



f

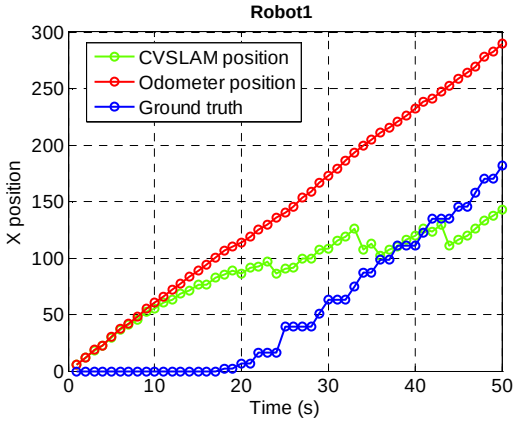
Figure 6. 9 Feature extraction and matching, a, b) features extracted in left and right image observed by robot1, c) Features correspondence for robot1; d,e) features extracted in left and right image observed by robot2, f) Feature correspondence for robot2, shared region observed by robot 1 and 2 are limited by the red line.

6.5.3 Experimental Navigation Results without Map Management

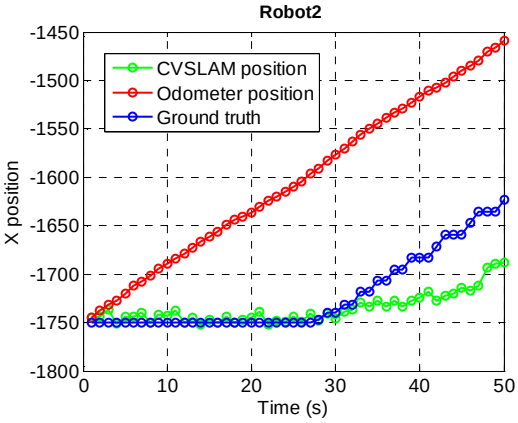
In the first part of these experiments, the CVSLAM without map management is validated. In this case, all observed features will be added to the state vector. Thus, the correlation between features and UAVs will increase, which will lead to an accurate estimation of the map and UAVs positions. However, the size of the state vector will increase linearly with time as shown in Figure (5.16). This is not suitable for long term navigation as it will impact on computation requirements and time.

6.5.3.1 Experimental Indoor Results

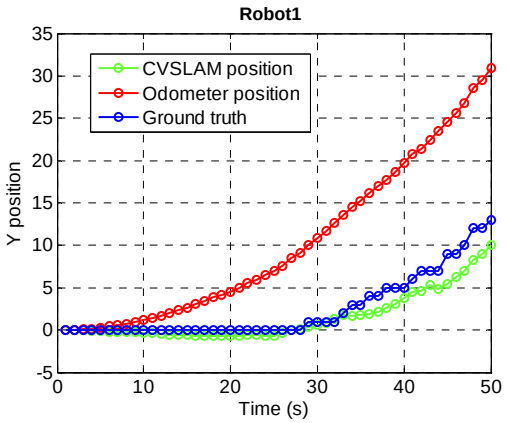
Figure (6.8.b, c) shows the state variables (X, Y and θ) of the two mobile robots considered in this experiment. Our vehicle has three degree of freedom rather than six as for aerial vehicles. However, the observation system is still the same as presented in Section 5.4, Chapter 5.



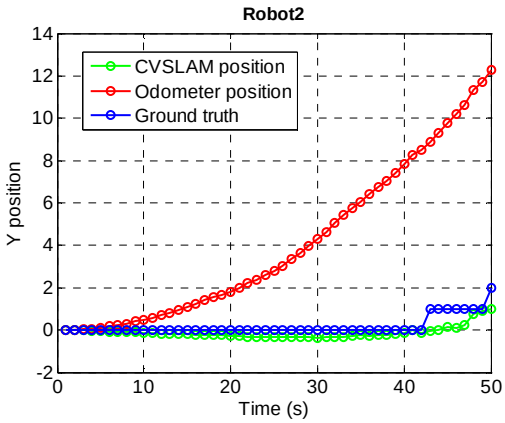
-a-



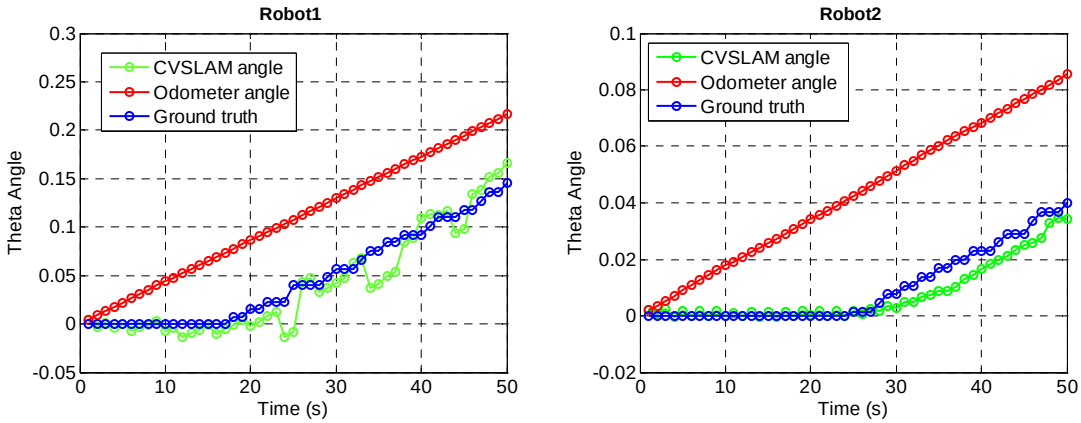
-d-



-b-



-e-

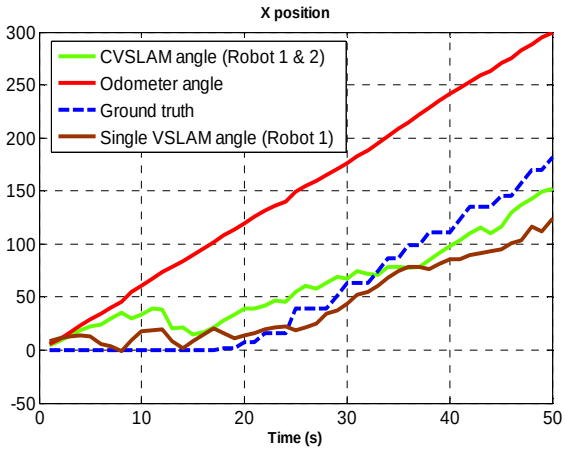


-c- -f-

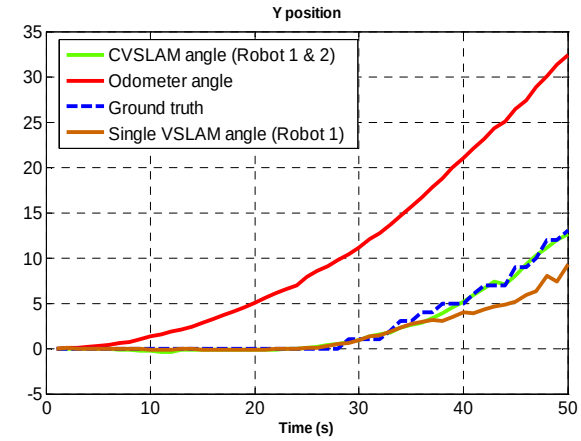
Figure 6. 10 Robots positions, left XYZ position of Robot1 in navigation frame Right XYZ position of Robot2 in navigation frame

Figure (6.10) shows the results of pose estimation using experimental data of two mobiles robot navigating in indoor environment. Each robot acquires images using the bumblebee stereo cameras Figure (6.8.a). As can be seen good pose estimation (position x, y and orientation θ) is obtained by the C-VSLAM algorithm comparing to the odometer pose.

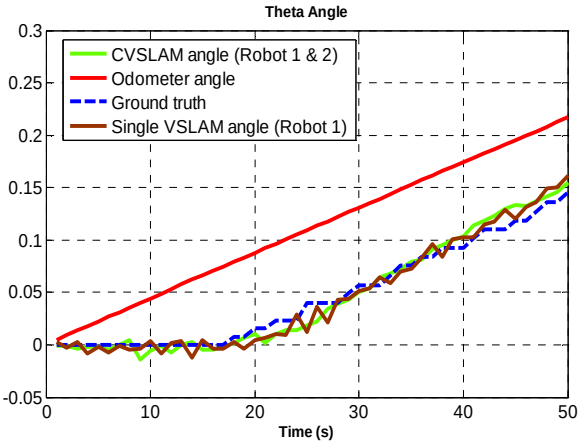
Comparison between Single VSLAM (S-VSLAM) and Cooperative VSLAM using real data is given in Figure (6.11). As can be seen from Figure (6.11.a and b) in short term navigation positions (X and Y) estimated by S-VSLAM and C-VSLAM are quite similar, however, when the navigation time increases the S-VSLAM estimator becomes less accurate comparing to the C-VSLAM estimator which maintains a good precision.



-a-



-b-



-c-

Figure 6. 11 Mobile robot localisation Single VSLAM V vs Cooperative VSLAM

Experimental results of the images mosaic using Cooperative VSLAM algorithm are presented in Figure (6.12, 13 and 14). Figure (6.12) and Figure (6.13) show the image mosaic constructed using mobile robot 1 and 2 respectively, while Figure (6.14) shows the image mosaic constructed using both robots. It is clear that cooperating robots build a larger map than each robot alone, which is a very important aspect when exploring large area of environment.

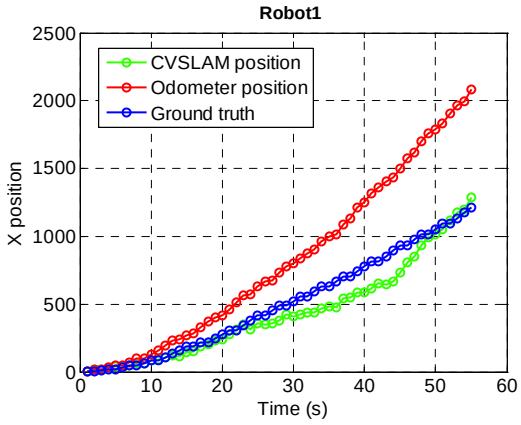
Figure 6. 12 Image mosaic constructed by Robot 1

Figure 6. 13 Image mosaic constructed by Robot 2

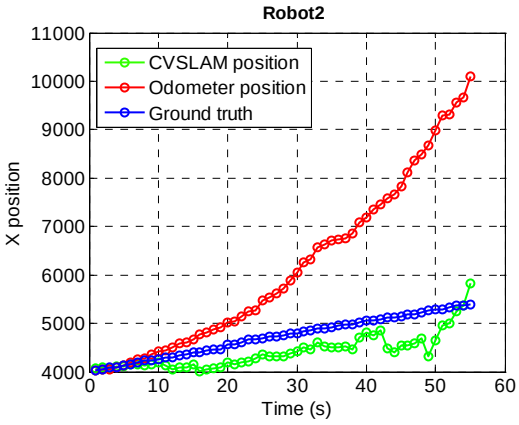
Figure 6. 14 Image mosaic constructed by Robot 1 and 2

6.5.3.2 Experimental Outdoor Results

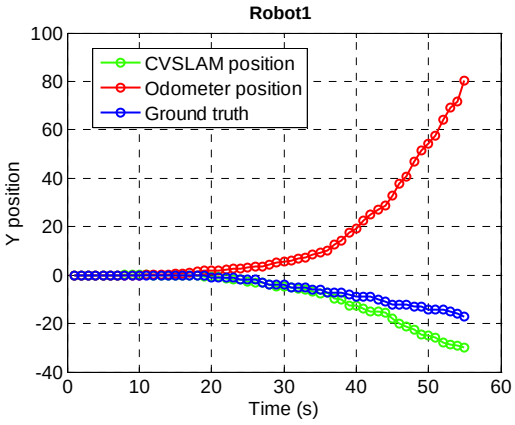
Experimental outdoor validation was also conducted in order to verify the robustness of our approach in a scenario when environment conditions are not controlled. Figure (6.15) shows the results of pose estimation using experimental data of two mobiles robot navigating in outdoor environment. As can be seen good pose estimation (position x, y and orientation θ) is obtained by the C-VSLAM algorithm comparing to the odometer pose.



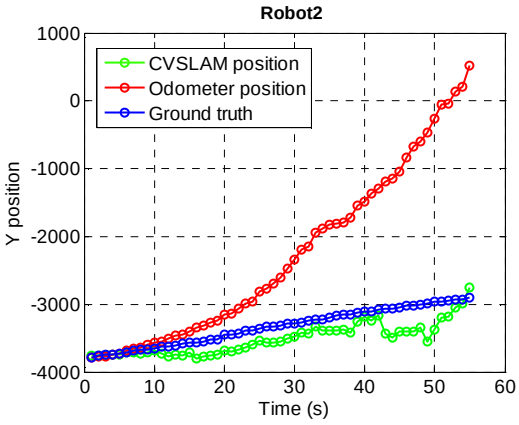
-a-



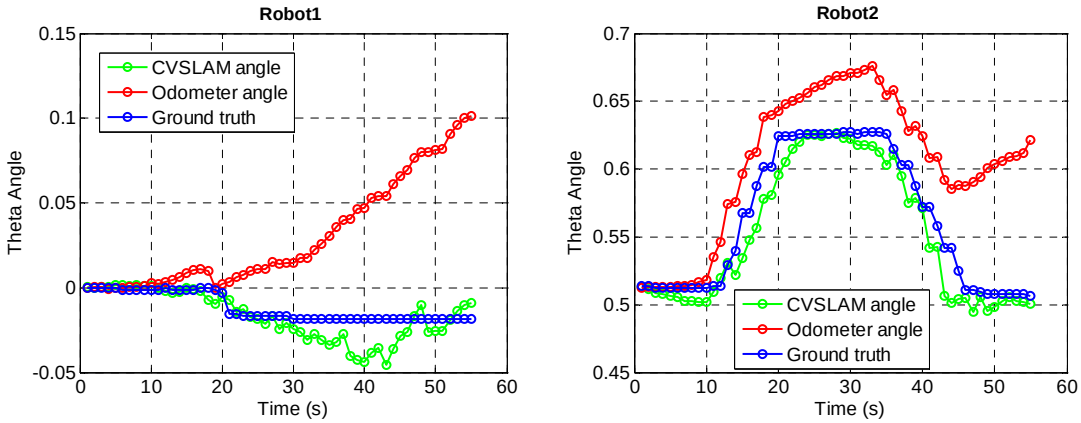
-d-



-b-



-e-



-c- -f-

Figure 6. 15 Robots positions, left XYZ position of Robot1 in navigation frame Right XYZ position of Robot2 in navigation frame, outdoor experience

Experimental results of the images mosaic using C-VSLAM algorithm in outdoor environment are presented in Figure (6.16, 17 and 18). Figures (6.16 and 6.17) show the image mosaics built using mobile robot 1 and 2 respectively. Figure (6.18) shows the image mosaic built in the command computer using both robot scene perceptions. From these figures, it is clear that cooperating robots will have access to a larger map than each robot alone, which is very important to explore large area in outdoor environments.

Figure 6. 16 Image mosaic constructed by Robot 1

Figure 6. 17 Image mosaic constructed by Robot 2

Figure 6. 18 Image mosaic constructed by Robot 1 and 2

As we mentioned in this section, without map management the $O(N^2)$ complexity of the H∞ filter (N being the dimension of the state vector) does not allow large environments to be efficiently mapped since it limits the total number of landmarks that can be stored in the map [90]. Beyond this upper limit, real-time processing is no longer possible [105]. To prevent the state vector from a rapid dimension increase that would dramatically limit the mapping capacity of our VSLAM system, we extend the map management approach presented in Chapter 5, Section (5.8) to the CVSLAM case.

6.5.4 Experimental Navigation Results with Map Management

At each time step, the state vector is proposed to contain the new observed features and the best k -landmarks observed previously by each UAV. As explained in

Section (5.8), this will help to maintain a suitable state vector size, building a large map and to make the loop closing detection possible with the few maintained features.

6.5.4.1 Single VSLAM with Map Management and Loop Closing

Figure (6.19) shows the pose estimation (position x, y and orientation θ) of a mobile robot obtained by the Single VSLAM with map management. As can be seen the S-VSLAM position is more accurate than the odometer position. However, S-VSLAM position diverges with time until ***t=70s*** when a loop closing is detected which increases the estimation accuracy. Map management positive is apparent here since that even with many features removed from the state vector (to reduce the CVSLAM complexity), the loop closing is well detected and the estimation of vehicle positions is fine.

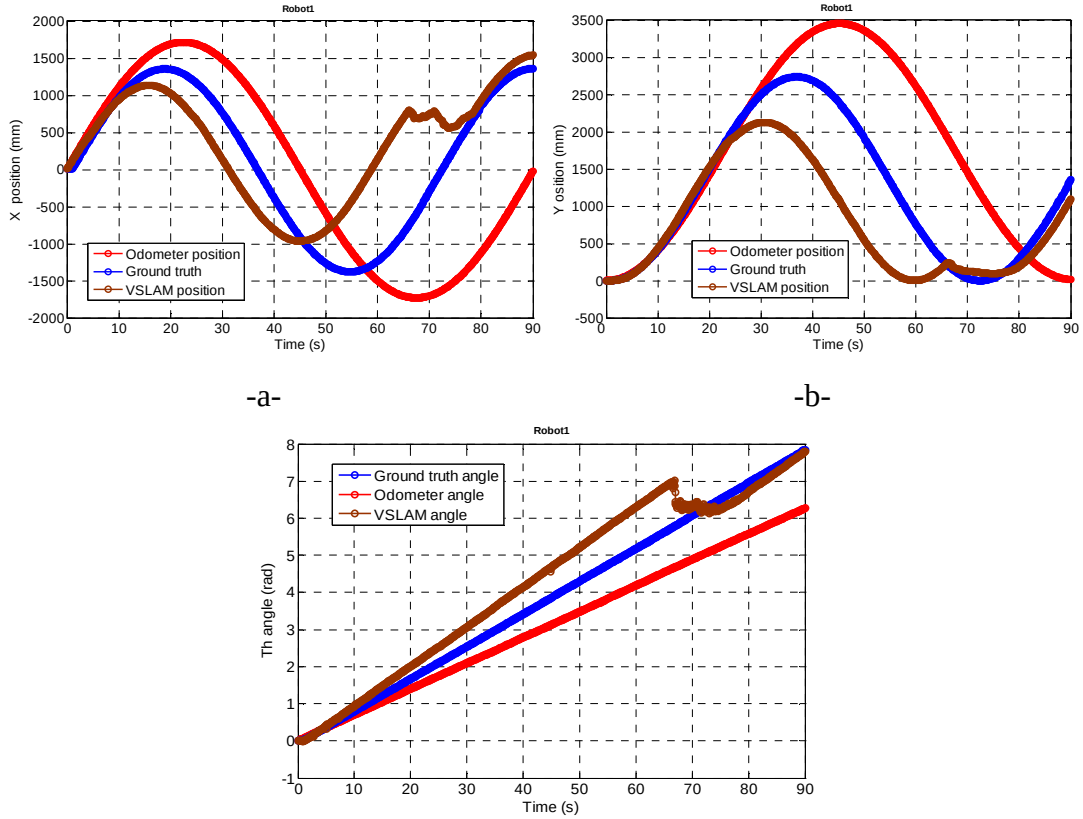


Figure 6. 19 Position of Robot1 using Single VSLAM with map management and loop closing

Figure (6.20) presents the interface that is built and run centrally on our command computer and based on the communication data transmitted by single robot. Displays of

trajectory estimation, of the detected features from the scene and of the good and false matching for the robot are given.

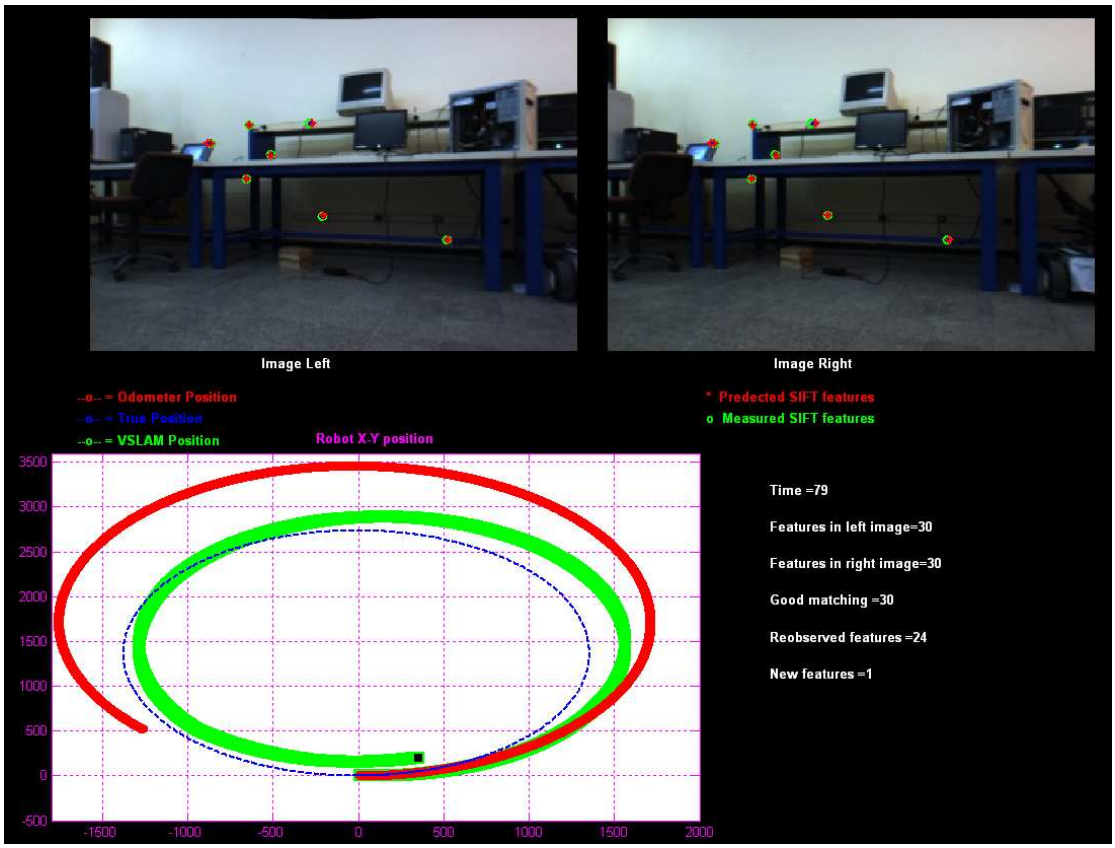
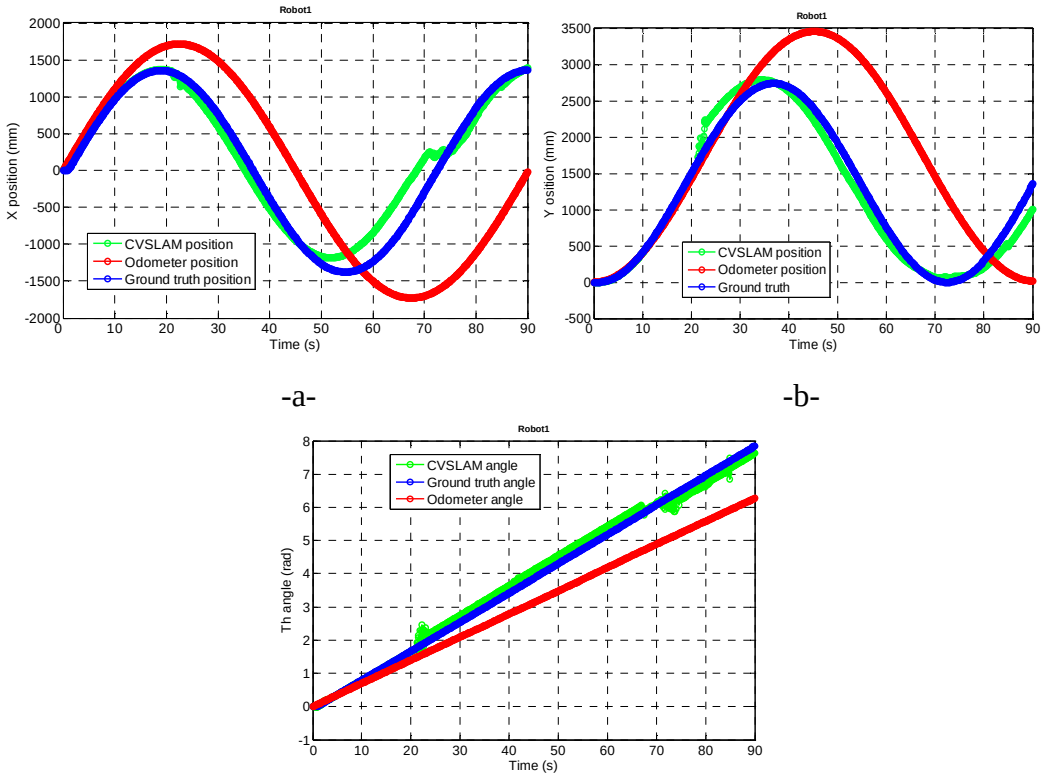


Figure 6. 20 Robot1 X-Y position using Single VSLAM with map management and loop closing

6.5.4.2 Cooperative VSLAM with Map Management and Loop Closing

Figures (6.21 and 22) show the results of the experimental pose estimation of the two cooperative robots navigating indoor. Good pose estimation (position x, y and orientation Theta) is obtained by the C-VSLAM algorithm comparing to the odometer pose for the two robots.

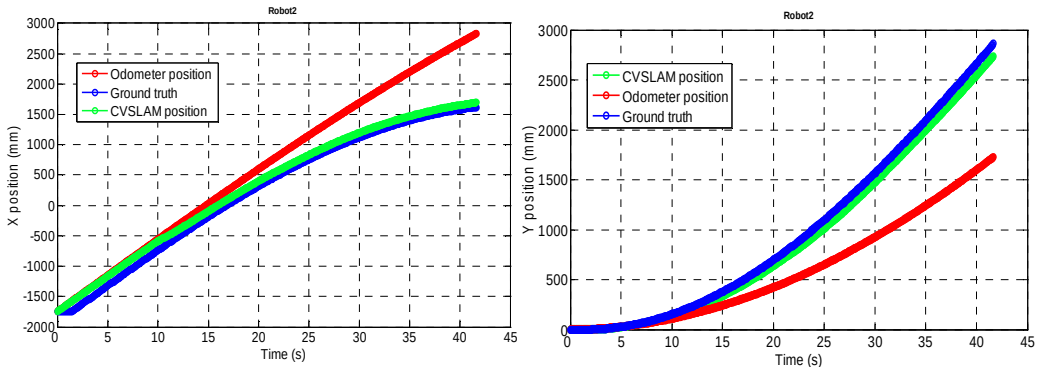
The advantages of the map management appear clearly from these results. Not only the CVSLAM complexity is reduced but also the loop closing is well detected for Robot 1 (although many features are removed from the state vector).



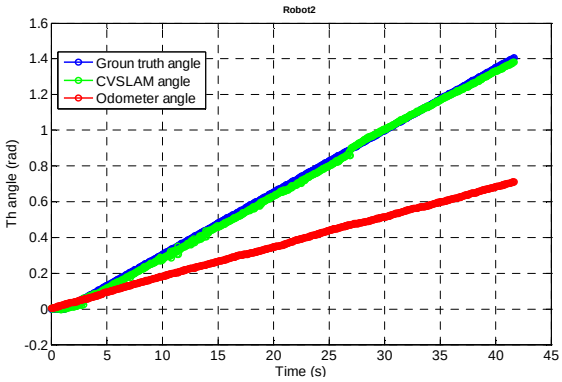
-a- -b-

-c-

Figure 6. 21 Position of Robot1 using cooperative VSLAM with map management and loop closing



-a- -b-



-C-

Figure 6. 22 Position of Robot2 using cooperative VSLAM with map management and loop closing

Figure (6.23) presents the interface that is built and run centrally on our command computer and based on the communication data transmitted by the two cooperative robots. Displays of trajectory estimation, of the detected features from the scene and of the good and false matching for the two robots are given.

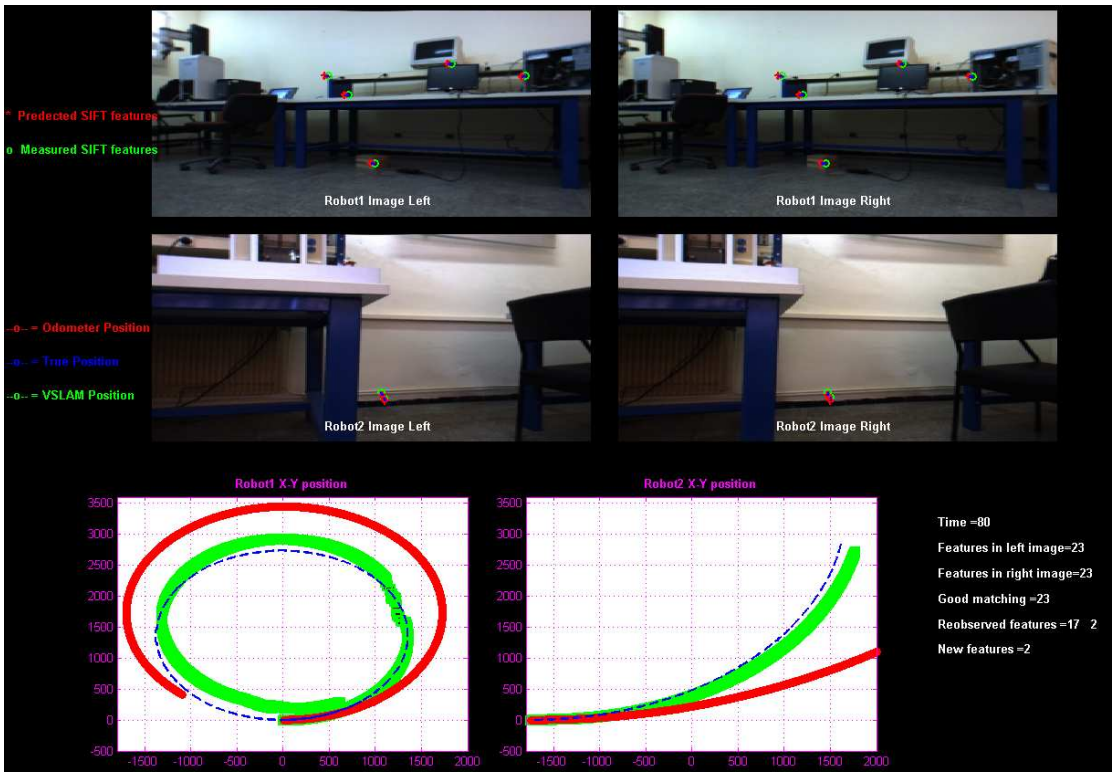


Figure 6. 23 Robot 1 and 2 X-Y position using Cooperative VSLAM with map management and loop closing

6.5.5 Single VSLAM vs Cooperative VSLAM

Comparison between experimental S-VSLAM and C-VSLAM is given in Figure (6.24). As it is shown, navigation positions (X and Y) estimated by S-VSLAM is less accurate comparing to the C-VSLAM estimates that maintain a good precision. In fact a zoom on the first instants of robot travels shows that C-VSLAM and S-VSLAM look close to each others. However, after a little travelling C-VSLAM navigation results outperform single navigation results. This is mainly due to the fact that to the sharing of visual features which augment the reliability and robustness of the estimation process by reducing perception uncertainties. In addition to this, loop closing effects is apparent on both single and cooperative VSLAM by looking carefully on Figure (6. 24) at time 70s.

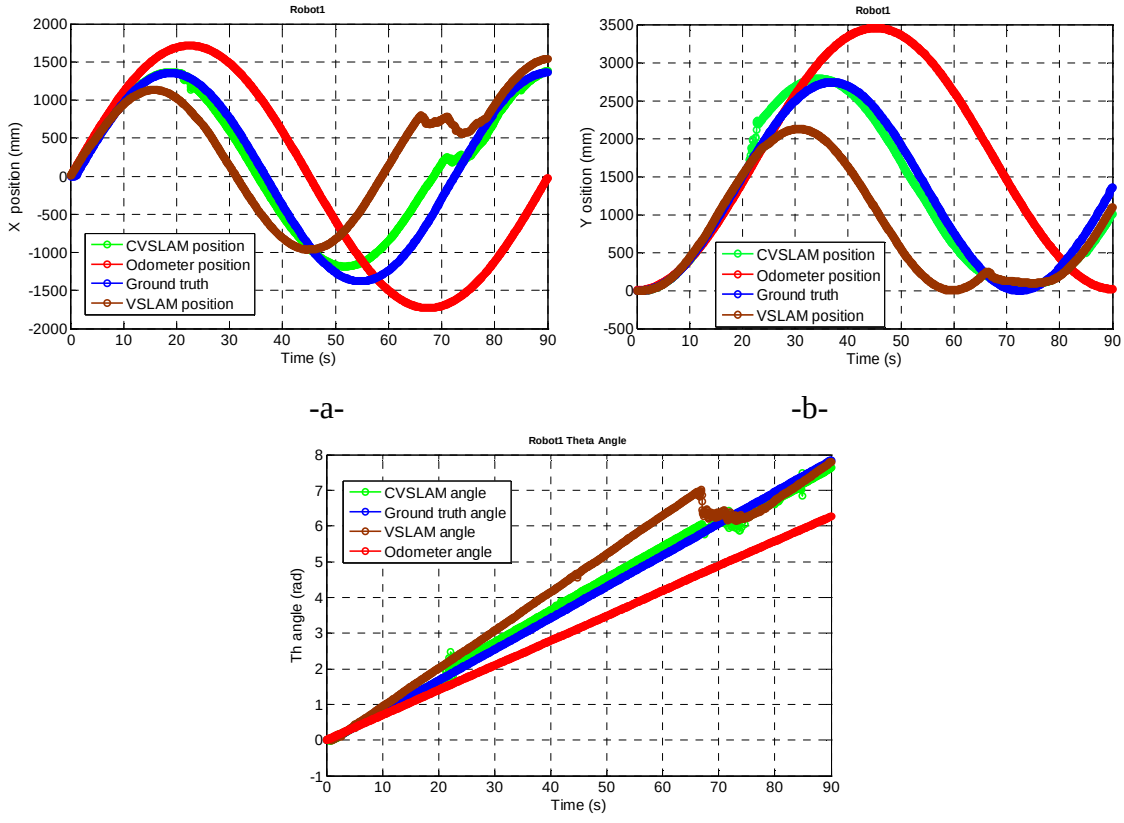


Figure 6. 24 Mobile robot localisation Single VSLAM V vs Cooperative VSLAM With map management and loop closing detection

6.6 Decentralised Cooperative VSLAM (DC-VSLAM)

The main advantage of the Centralised Cooperative VSLAM is the construction of a reliable large map and the estimation of an accurate UAV position. However, the centralised architecture has two main drawbacks. First, failure of one component results in complete failure of the whole system. Second, UAVs are not fully autonomous but depend on the ground station, Figure (6.1). This makes UAVs able to navigate in a limited area around the ground station only because of communication and delays constraints. One promising solution to overcome these constraints is to propose a Decentralised Cooperation of UAVs. In this latter each UAV has its own S-VSLAM algorithm and the cooperation between UAVs will be taken in consideration only when shared features are observed. Many works are proposed in the literature to solve the Decentralised CVSLAM problem. Most of them have been devoted to the definition of different architectures [137-140]. Mostly behavior-based [141-142], that rule the interaction between the behaviors of individual robots.

6.6.1 UAVs Cooperation in DC-VSLAM

Communication is a central issue of multi vehicle system because it determines the possible modes of interaction among vehicles, as well as their ability to build successfully a world model. Communication may appear in three different forms of interaction [143]: (i) *via environment*, using the environment itself as the communication medium; (ii) *via sensing*, when an agent uses its sensing capabilities to observe and perceive the actions of its teammates; and (iii) *via communication*, using a communication channel to explicitly exchange messages among the agents to compensate perception limitations.

The proposed Decentralised Cooperative VSLAM (DC-VSLAM) will be mainly based on the third form of interaction which is via explicit communication. The most restrictive constraint for explicit communication is the limited amount of data to communicate between UAVs. As a result, the collective data to share should be selected carefully in order to maximise the gain with a minimum communication.

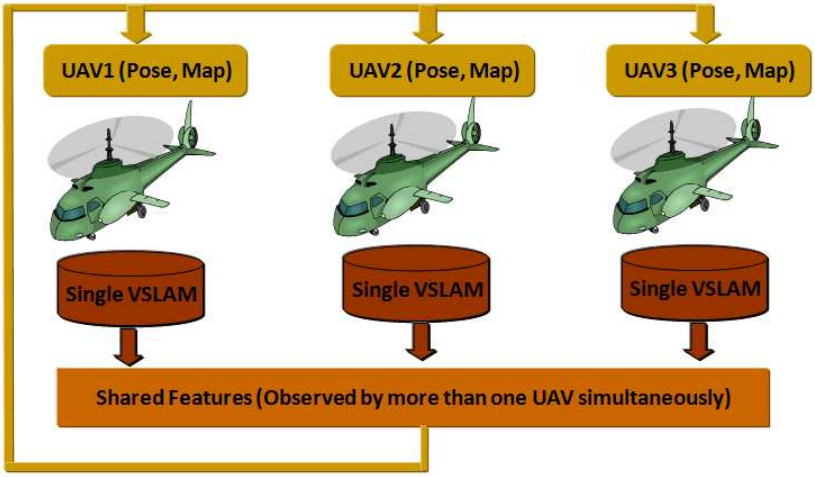


Figure 6. 25 Decentralised architecture

Figure (6.25) shows the architecture of the Decentralised C-VSLAM. Each UAV has its own S-VSLAM and when shared features are observed their positions and uncertainties are updated using Distributed Estimation (DE).

6.6.2 Which UAVs Will Cooperate?

Assume we have N UAVs ($UAV_1, UAV_2, ..., UAV_N$). At $t = t_i$ each UAV_i observes M_i feature $f^i(x^i, y^i, z^i)$. Our strategy is then to detect possible shared region is as follows:

For each UAV_i we calculate the mean (μ^i) and the standard deviation (σ^i) of the observed features f^i. Then, each set of observed features will be approximated by an ellipsoid (ξ^i) centered at (μ^i) with axes (σ^i). Therefore, UAV_i and UAV_j will cooperate ($Coop(i, j) = 1$) if and only if: $\xi^i \cap \xi^j \neq \phi$.

6.6.3 DC-VSLAM Strategy

The proposed Decentralised C-VSLAM strategy is defined as follows: assume we have three UAVs (UAV_i, UAV_j, UAV_k) observing a number of features (N_i, N_j, N_k) respectively. Each observed feature has a descriptor or an index.

The proposed strategy, Figure (6.26), allows communication between UAVs of: number of observed features, indexes of these features and the positions and uncertainties of shared features referred by the communicated indexes. Then, the

Covariance Intersection (CI) approach is used to estimate the position and covariance of shared features before adding them to the global map.

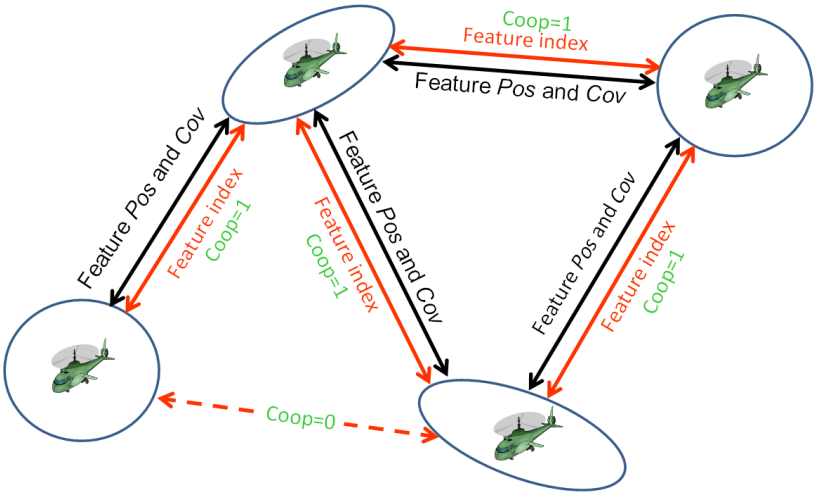


Figure 6. 26 DC-VSLAM strategy

6.6.4 Covariance Intersection

6.6.4.1 Introduction

Covariance Intersection (CI) enforces consistency of system state estimates by means of additional weighting coefficients [149]. In order to obtain variance estimates that are not far too conservative, the weighting coefficients should be chosen with a certain care. One would be able to produce less conservative estimates by, e. g., selecting the weights in accordance to a tight bound for the joint covariance [144] or using a split covariance intersection algorithm [145], respectively. But, these and other approaches to incorporate additional knowledge into the fusion process as, for example, discussed in [146-147] are beyond the scope of this work. In some applications, the implementation of such more involved approaches may not even be possible due to limited computing resources and/or low communication bandwidths. It is assumed from now on that a precise quantification of cross-correlation between estimates is not available.

With respect to the weighting coefficients for covariance intersection, it is usually suggested to minimize either the trace or the determinant of the resulting error variance matrix. In order to avoid the possibly high numerical effort for solving this nonlinear convex optimisation problem, an approximate Fast Covariance Intersection (FCI)

scheme has been proposed by Niehsen [148]. In due course, it will turn out that the coefficients obtained by FCI approach may deviate significantly from the solutions of the aforementioned optimisation problem [149].

6.6.4.2 Fast Covariance Intersection

Starting from two features estimates $\hat{x}_1$ and $\hat{x}_2$ of the true feature position x given by UAV_1 and UAV_2 respectively and the corresponding positive definite error variance matrices P_1 and P_2, a combined estimate $\hat{x}$ with error variance matrix P is sought. For uncorrelated estimates $\hat{x}_1$ and $\hat{x}_2$, the overall Minimum Mean Square Error (MMSE) estimate is given by basic convex combination of the two estimates.

$$\begin{cases} \hat{x} = PP_1^{-1}\hat{x}_1 + PP_2^{-1}\hat{x}_2 \\ P^{-1} = P_1^{-1} + P_2^{-1} \end{cases} \tag{6.1}$$

It is known that in presence of correlation between the two initial estimation errors, the estimated P may become far too optimistic. This can finally cause divergence in sequential filtering. Yet, without any detailed knowledge about the correlation, at least a conservative estimate can be given by applying covariance intersection according to:

$$\begin{cases} \hat{x} = \omega_1 PP_1^{-1}\hat{x}_1 + \omega_2 PP_2^{-1}\hat{x}_2 \\ P^{-1} = \omega_1 P_1^{-1} + \omega_2 P_2^{-1} \end{cases} \tag{6.2}$$

with nonnegative coefficients ω_1 and ω_2 obeying

$$\omega_1 + \omega_2 = 1$$

Herein, the coefficients ω_1 and ω_2 are usually chosen as to minimize either the trace or the determinant of P. In order to avoid high numerical computations and to find the solution of such a highly nonlinear optimisation problem, Niehsen has proposed to use a fast approximate solution instead. Niehsen has argued that a replacement of P_1 by P_2 and vice versa must lead to correspondingly switching coefficients ω_1 and ω_2. For $trace(P_1) << trace(P_2)$ one would expect to get $\omega_1 \approx 1$.

with that, Niehsen [148] has suggested to use the following solution:

$$\omega_1 trace(P_1) - \omega_2 trace(P_2) = 0 \tag{6.3}$$

$$\omega_1 = \frac{trace(P_2)}{trace(P_1) + trace(P_2)} \tag{6.4}$$

6.6.4.3 Adaptive Covariance Intersection

The previous equations of Fast Covariance Intersection (FCI) are useful for static sensors when sensor positions are known accurately. However, in our case the cameras are embedded on UAVs navigating in 6DoF and the positions of those UAVs are estimated with uncertainties. These latter are not taken in consideration in the Fast Covariance Intersection scheme proposed by Niehsen [148]. For example assume UAV_1 (UAV_2) position is estimated with covariance P_{UAV1} (P_{UAV2}) as illustrated in Figure (6.27). Then, if we assume $trace(P_{UAV1}) << trace(P_{UAV2})$, which mean UAV_1 position is estimated more accurately than UAV_2. In this case, feature estimation given by UAV_1 should have more weight than that given by UAV_2. As consequence:

when $trace(P_{UAV1}) << trace(P_{UAV2})$ then $\omega_1 \to 1$, $\omega_2 \to 0$ which means $\hat{x} \to \hat{x}_1, P \to P_1$. From this analysis and from the Niehsen [148] analysis, we propose a new Adaptive Covariance Intersection scheme, where the determination of ω_1 depends not only to $trace(P_1)$ and $trace(P_2)$ but also by $trace(P_{UAV1})$ and $trace(P_{UAV2})$. Thus as it is mentioned in Figure (6.27):

$\omega_1 \to 1$ when $trace(P_{UAV1}) + trace(P_1) << trace(P_{UAV2}) + trace(P_2)$, as a result Equation (6.4) will be formulated as follows:

$$\omega_1 = \frac{trace(P_2) + trace(P_{UAV2})}{trace(P_1) + trace(P_2) + trace(P_{UAV1}) + trace(P_{UAV2})} \quad \textbf{(6.5)}$$

This manner to calculate ω_1 is more suitable and realistic for DC-VSLAM problem when neither UAV positions are known nor feature positions are known.

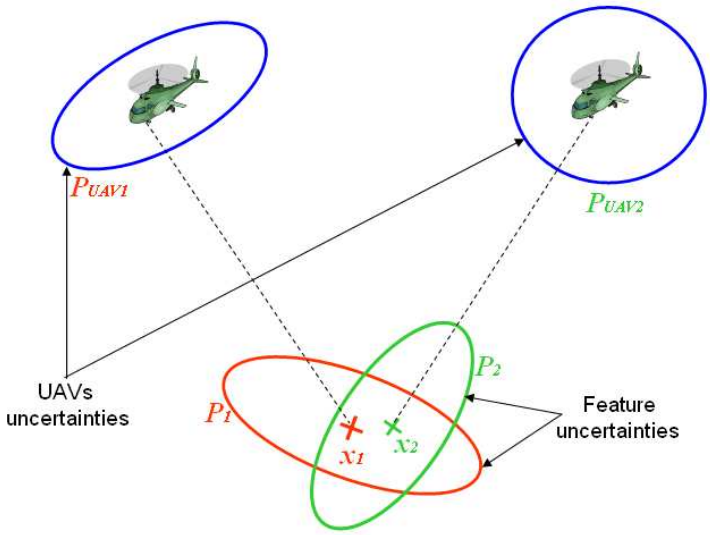


Figure 6. 27 Adaptive Covariance Intersection

6.6.5 DC-VSLAM algorithm

Our DC-VSLAM algorithm is based on taking into consideration the limited amount of data to exchange between UAVs. During navigation, each UAV observes a set of features. Each feature has a descriptor or an Index. Then, the mean and standard deviation of this set of features is calculated and approximated by an ellipsoid for each UAV. Two UAVs will cooperate if and only if there is an intersection between feature ellipsoids. If this condition is satisfied then, Indexes will be transmitted from the UAV that is observing fewer features to the second UAV in order to check which features are shared (have similar Index). After that, shared features are communicated. Adaptive Covariance Intersection (ACI) will then run in the UAV receiving features to estimate the position and covariance of shared features before adding them to the global map. The main steps of DC-VSLAM algorithm are presented below:

At each step time t

- *If* $Coop(i, j) = 1$
  - N_i *number of features observed by UAVi*
  - N_j *number of features observed by UAVj*
    - *If* $N_i < N_j$
      - N_i *Indexes* $[Ind(i,1)...Ind(i, N_i)]$ *will be transmitted to* UAV_j
        - *For l=1:* N_i
          - *If* $Ind(i,l) \in [Ind(j,1)...Ind(j, N_i)]$
            - $Ind(i,l)$ is a shared feature.
            - $Ind(i,l)$ will be transmitted to UAV_i
            - $Pos(Ind(i,l))$ *and* $Cov(Ind(i,l))$ *will be transmitted to* UAV_j
            - Adaptive Covariance Intersection (ACI) will be used to estimate the position and covariance of feature observed by UAV_i *and* UAV_j
          - *End*
        - *End*
      - *Else*
        - *The same algorithm will run in* UAV_i
    - *End*
  - *End*

End

6.6.6 CC-VSLAM vs DC-VSLAM

In the Centralised Cooperative VSLAM (CC-VSLAM), the state vector contains UAV positions and all observed features. This way, the correlation between UAV-UAV, UAV-Feature and Feature-Feature can be estimated. This leads to an accurate state estimation. However, such strategy can result with time in a state vector size increase, which is not suitable for long term navigation.

On the other hand, in Decentralised Cooperative VSLAM (DC-VSLAM) each UAV has its own state vector. The cooperation between UAVs is done only when shared features are available. Otherwise, each UAV is considered to run its Single VSLAM. Thus, DC-VSLAM will result in a similar performance or less than the one obtained by using CC-VSLAM. Positive effects of DC-VSLAM are more in its flexibility and suitability to deal with restricted communication, system complexity and timely decision making for embedded systems and real time applications.

6.6.7 Simulation results

Figure (6.28) shows the results of simulation of two cooperative UAVs using the Decentralised VSLAM approach. The green UAV position is estimated accurately with DC-VSLAM even without loop closing detection. The shared regions between the two UAVs (dashed pink ellipses) were enough to reach a suitable accuracy.

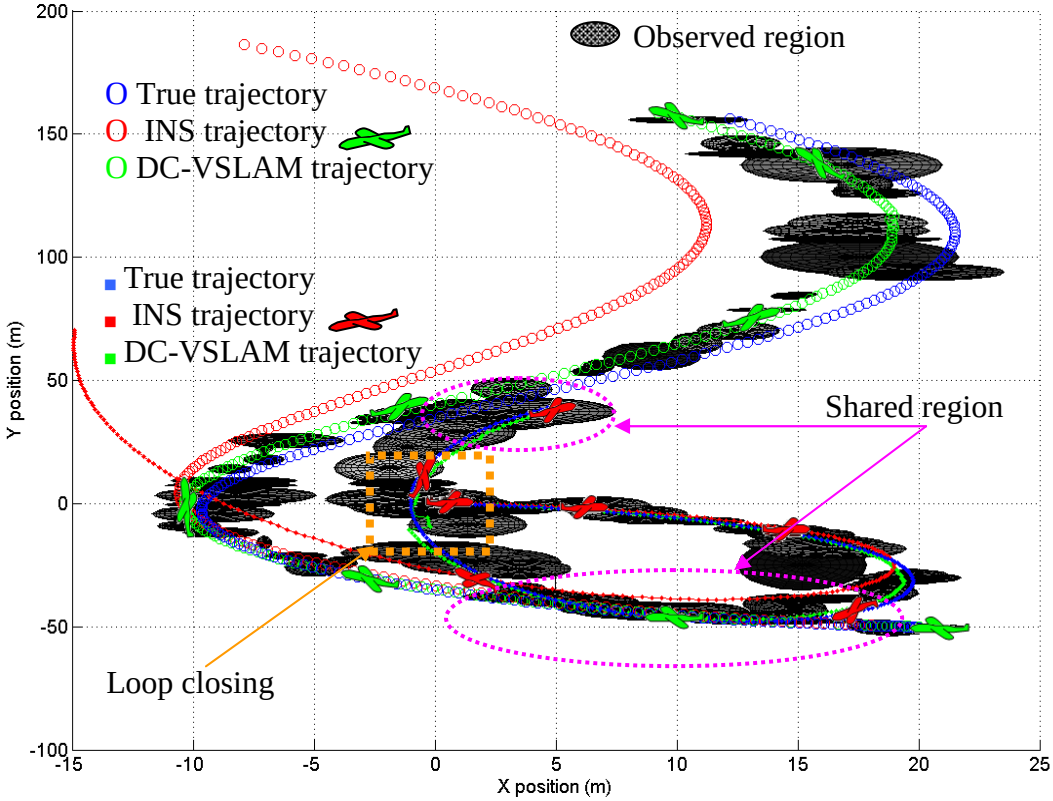


Figure 6. 28 Two UAVs cooperating using the DC-VSLAM

XYZ positions in navigation frame and UVW velocities in body frame of the UAV1 (red) and UAV2 (green) are given in Figure (6.29 and 6.30) respectively. From

the figures good estimations are obtained for X, Y and Z positions for both UAVs. From Figure (6.29.f), significant decrease of the estimation error is observed at ***t=200s*** when loop closing is detected (UAV1 re-visits the same area, dashed square in Figure (6.28)). From Figure (6.30), good pose estimation is obtained for UAV2 even without making a loop. This can be explained by the efficiency of DC-VSLAM algorithm in exploiting the shared regions between UAVs.

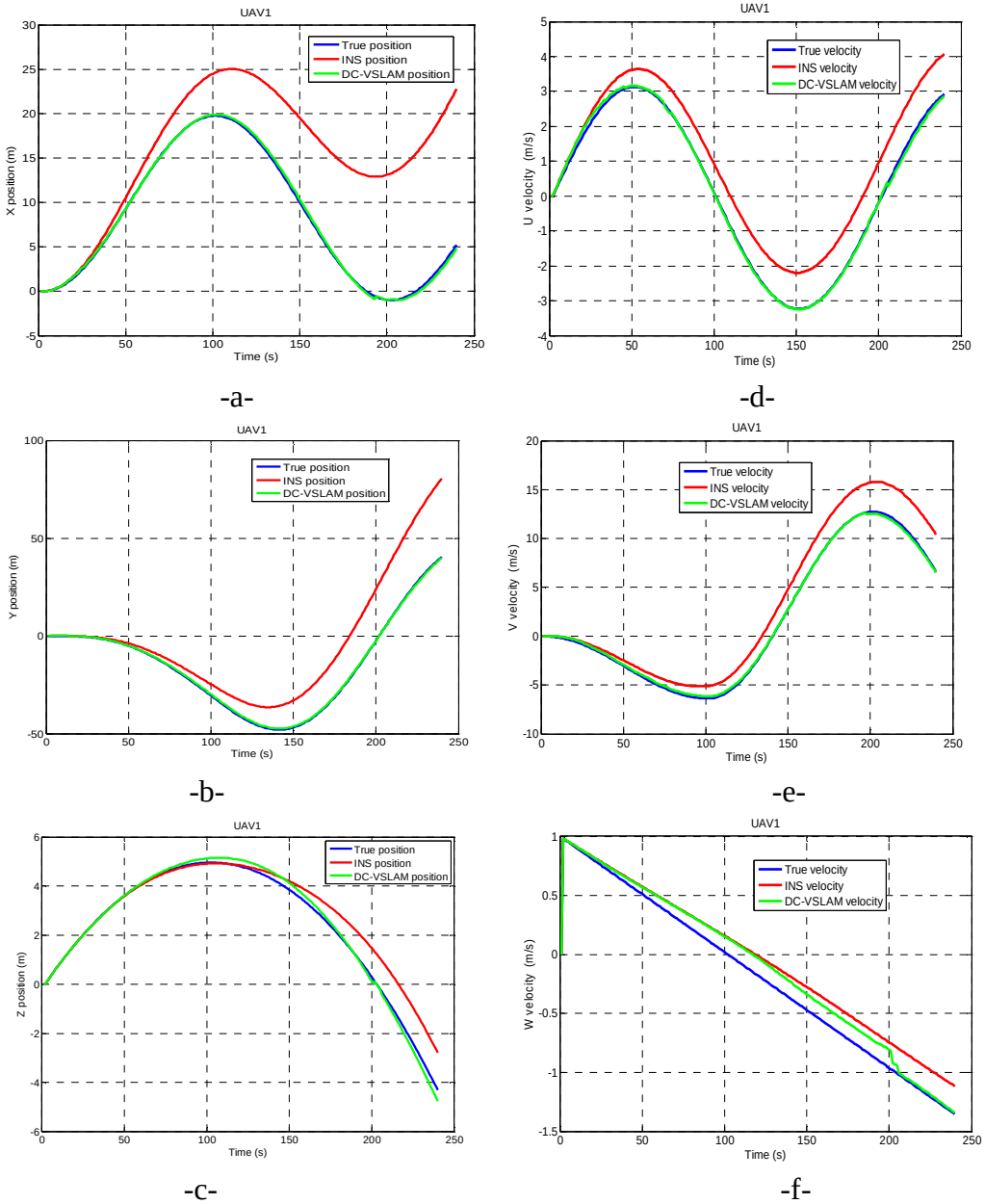


Figure 6. 29 UAV1 positions and velocities, left XYZ position in navigation frame Right UVW velocity in body frame

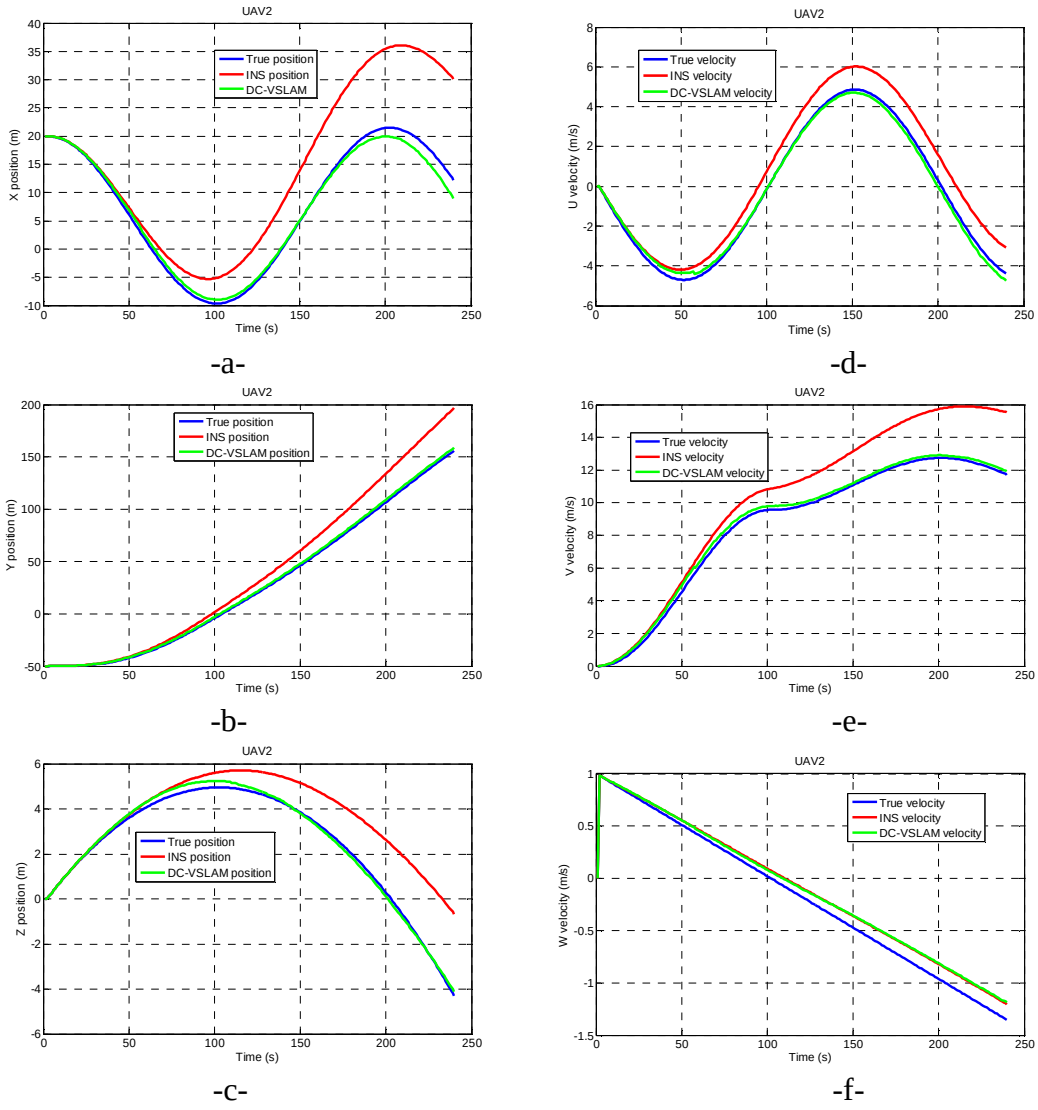


Figure 6. 30 UAV2 positions and velocities, left XYZ position in navigation frame Right UVW velocity in body frame

If we compare these results with those obtained by the CC-VSLAM, we can conclude that quite similar performances are obtained. Both results achieve good pose estimation with loop closing detection. However, DC-VSLAM is more suitable for embedded systems and real time application as stated in the section above.

6.7 Conclusion

In this chapter, we proposed a robust approach to solve the cooperative Airborne VSLAM problem based on the development of a full stereo camera observation model. An adaptive SIFT feature extractor followed by stereo vision constraints fast matching

were introduced to construct a large map. Robust Cooperative VSLAM is implemented based on the Nonlinear H∞ filter and compared with the Single VSLAM. The proposed Centralised Cooperative VSLAM solution was validated using simulation and experimental data and good and very promising results were obtained. In the last section of this chapter we proposed another alternative to solve the Cooperative VSLAM based on a decentralised architecture. This latter is more suitable for embedded applications. The adaptive DC-VSLAM strategy is validated in simulation and promising results were obtained.

Chapter 7

Conclusion and future work

CHAPTER 7

Conclusions and Future work

The aim of this thesis is to achieve the UAV full autonomy by developing tools able to provide an accurate automatic localisation in an accurate environment map. SLAM problem looks at the ability of an autonomous vehicle, starting in a partially known or unknown environment, to incrementally build an environment map and simultaneously localise itself within this map. New challenges in SLAM are delivering methods enabling large-scale implementations in increasingly unstructured environments and especially in situations where GPS-like solutions are unavailable or unreliable such as urban canyons and urban environments.

This thesis has studied a number of technical problems that are necessary to solve in order to increase UAV autonomy. First, UAV localisation has been investigated. Then, UAV map building was presented followed by an implementation of a simultaneous localisation and mapping solution using visual observation. Finally, we extended the single UAV VSLAM solution to the cooperative VSLAM case. The presented work has been virtually divided into four integrated contributions. First, in Chapter (3), a proof of SDRE stability has been developed and validated using simulation data. Next, INS/GPS UAV localisation techniques using EKF/UKF/SDRE/ $NH\infty$ have been proposed, implemented and compared using experimental data. Second, an Adaptive Scale Feature Transform (ASIFT) was proposed in Chapter (4) to detect suitable number of features for any kind of airborne visual environment. ASIFT was validated in image mosaic construction using real images. In the same chapter a new feature detector/descriptor Scale Wavelet Invariant Feature (SWIF) was introduced more specifically for VSLAM and real time imaging application. Third, in Chapter (5) the 3D VSLAM problem has been solved using the robust nonlinear $H\infty$ filtering scheme. This latter does not make any assumption about noise characteristics and is robust against process and observation modeling errors.

Observability and consistency analysis are investigated in this chapter. A novel map management approach is also proposed in order to make the algorithm suitable for real time implementation. Experiments using real data have illustrated the robustness of $NH\infty$ filter when noise is not centred or not white. Finally, in chapter (6), Centralised and Decentralised Cooperative VSLAM solutions have been developed and presented. Experiments using real data have demonstrated the efficiency of the proposed methods to construct a large map of the environment and to estimate well positions of autonomous vehicles within this map.

Future work

As every research work, this thesis has been limited by time and funds. Thus, a number of research challenges have been identified and proposed for future investigation. In chapter (3), contributing at solving the INS/GPS localisation problem by taking into account system uncertainties to increase the navigation solution robustness is a challenging problem. Another area of research that can be explored following the work presented in Chapter (4) is the application and validation of the new SWIF Detector/Descriptor to VSLAM real time application. In Chapter (5), an intuitive suggestion of future work is to solve the airborne VSLAM problem using SDRE filtering technique which presents a major advantage since it avoids linearisation problems**.** In Chapter (6), Cooperative VSLAM as new subject of research can be explored and investigated further and more deeply. For Decentralised CVSLAM scheme, many ideas and contributions could be thought of. An optimal architecture of communication and data sharing between a large numbers of UAVs is of crucial importance to this decentralised cooperative scheme. Extending our proposition in Chapter (6) and Validating it using a team of UAVs with embedded VSLAM for each of the UAVs is a challenging future task.

Furthermore, one important aspect to be investigated in the future is the real time constraints (algorithms complexity and computation time). This latter is very important to select the suitable algorithm for real time implementation.

Bibliography

BIBLIOGRAPHY

[1] R. Smith and P. Cheesman, “On the representation of spatial uncertainty”, Int. J. Robot. Res., vol. 5, no. 4, pp. 56–68, 1987.

[2] H.F. Durrant-Whyte, “Uncertain geometry in robotics”,IEEE Trans. Robot. Automat., vol. 4, no. 1, pp. 23–31, 1988.

[3] N. Ayache and O. Faugeras, “Building, registrating, and fusing noisy visual maps”, Int. J. Robot. Res., vol. 7, no. 6, pp. 45–5, 1988.

[4] J. Crowley, “World modeling and position estimation for a mobile robot using ultra-sonic ranging”, in Proc. IEEE Int. Conf. Robot. Automat, 1989, pp. 674–681.

[5] R. Chatila and J.P. Laumond “Position referencing and consistent world modeling for mobile robots”, in Proc. IEEE Int. Conf. Robot Automat, 1985, pp. 138–143.

[6] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics”, in Autonomous Robot Vehicles, I.J. Cox and G.T. Wilfon, Eds. New York: Springer-Verlag, pp. 167–193, 1990.

[7] J.J. Leonard and H.F. Durrant-Whyte, “Simultaneous map building and localisation for an autonomous mobile robot”, in Proc. IEEE Int. Workshop Intell. Robots Syst. (IROS), Osaka, Japan, 1991, pp. 1442–1447.

[8] J.J. Leonard and H.F. Durrant-Whyte, “Directed Sonar Navigation. Norwell”, MA: Kluwer, 1992.

[9] W.D. Renken, “Concurrent localisation and map building for mobile robots using ultrasonic sensors”, in Proc. IEEE Int. Workshop Intell. Robots Syst. (IROS), 1993.

[10] H. Durrant-Whyte, D. Rye, and E. Nebot, “Localisation of automatic guided vehicles”, in Robotics Research: The 7th International Symposium (ISRR’95), G. Giralt and G. Hirzinger, Eds. New York: Springer Verlag, pp. 613–625, 1996.

[11] M. Csorba, “Simultaneous Localisation and Map Building”, Ph.D. dissertation, Univ. Oxford, 1997.

[12] M. Csorba and H.F. Durrant-Whyte, “A new approach to simultaneous localisation and map building”, in Proc. SPIE Aerosense, Orlando, FL, 1996.

[13] J.J. Leonard and H.J.S. Feder, “A computational efficient method for large-scale concurrent mapping and localisation”, in Robotics Research, The Ninth International Symposium (ISRR’99), J. Hollerbach and D. Koditscheck, Eds. New York: Springer-Verlag, pp. 169–176, 2000.

[14] J.A. Castellanos, J.M. Martnez, J. Neira, and J.D. Tardós, “Experiments in multisensor mobile robot localisation and map building”, in Proc. 3rd IFAC Sym. Intell. Auton. Vehicles, 1998, pp. 173–178.

[15] J.A. Castellanos, J.D. Tardós, and G. Schmidt, “Building a global map of the environment of a mobile robot: The importance of correlations”, in Proc. IEEE Int. Conf. Robot. Automat, 1997, pp. 1053–1059.

[16] J. Guivant, E.M. Nebot, and S. Baiker, “Localisation and map building using laser range sensors in outdoor applications”, J. Robot. Syst., vol. 17, no. 10, pp. 565–583, 2000.

[17] S.B. Williams, P. Newman, G. Dissanayake, and H.F. Durrant-Whyte, “Autonomous underwater simultaneous localisation and map building” in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), San Francisco, CA, Apr. 2000, pp. 1793–1798.

[18] K.S. Chong and L. Kleeman, “Feature-based mapping in real, large scale environments using an ultrasonic array”, Int. J. Robot. Res., vol. 18, no. 1, pp. 3–19, 1999.

[19] M. Deans and M. Hebert, “Experimental comparison of techniques for localisation and mapping using a bearing-only sensor” in Proc. Int. Symp. Experimental Robot., 2000, pp. 395–404.

[20] J. Hollerbach and D. Koditscheck, Eds. Robotics Research, the Ninth International Symposium (ISRR’99). New York: Springer-Verlag, 2000.

[21] S. Thrun, D. Fox, and W. Burgard, “A probabilistic approach to concurrent mapping and localisation for mobile robots”, Mach. Learning, vol. 31, no. 1, pp. 29–53, 1998.

[22] Jong-Hyuk Kim, Salah Sukkarieh, "Real-time implementation of airborne inertial-SLAM", Robotics and Autonomous Systems 55(1): 62-71 (2007).

[23] Sunghwan Ahn, Minyong Choi, Jinwoo Choi and Wan Kyun Chung “Data Association Using Visual Object Recognition for EKF-SLAM in Home Environment” Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems October 9 - 15, 2006, Beijing, China.

[24] Jungho Kim, Kuk-Jin Yoon, Jun-Sik Kim and Inso Kweon, “Visual SLAM by Single-Camera Catadioptric Stereo SICE-ICASE” International Joint Conference 2006 Oct. 18-21, 2006 in Bexco, Busan, Korea.

[25] Woo Yeon Jeong Kyoung Mu Lee "Visual SLAM with Line and Corner Features", Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems October 9 - 15, 2006, Beijing, China.

[26] Pedro Pini´es, Todd Lupton, Salah Sukkarieh, Juan D. Tard´os, "Inertial Aiding of Inverse Depth SLAM using a Monocular Camera", ICRA 2007: 2797-2802.

[27] David Schleicher, Luis M. Bergasa, Rafael Barea, Elena López, Manuel Ocaña. " Real-Time Simultaneous Localisation and Mapping Using a Wide-Angle Stereo Camera", Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06).

[28] Andrew J. Davison, Ian Reid, Nicholas Molton and Olivier "Stasse MonoSLAM: Real-Time Single Camera SLAM", IEEE Trans. PAMI 2007.

[29] Javier Civera, Andrew J. Davison and J. M. M. Montiel "Inverse Depth to Depth Conversion for Monocular SLAM" , ICRA 2007.

[30] Jong-Hyuk Kim, Salah Sukkarieh: "Towards Robust Airborne SLAM in Unknown Wind Environments"

[31] Jong-Hyuk Kim, Salah Sukkarieh "Robust Multi-loop Airborne SLAM in Unknown Wind Environments", Proceedings of the 2006 IEEE International Conference on Robotics and Automation Orlando, Florida - May 2006.

[32] R. Greenspan "GPS and Inertial Navigation", chapter 7, American Institute of Aeronautics and Astronautics, Inc, 1996.

[33] J. Kim and S. Sukkarieh, "Flight Test Results of GPS/INS Navigation Loop for an Autonomous Unmanned Aerial Vehicle (UAV)", In International Technical

Meeting of the Satellite Division of the Institute of Navigation (ION), pages 510–517, Portland, OR, USA, September 2002.

[34] S. Snyder, B. Schipper, L. Vallot, N. Parker, and C. Spitzer, “Differential GPS/Inertial Navigation Approach/Landing Flight Test Results”. IEEE Transactions on Aerospace and Electronic Systems Magazine, pages 3–11, 1992.

[35] S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, “A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications”. IEEE Transactions on Automatic Control, 15:572–578, June 1999.

[36] Parkinson, B.W., and Spilker, J.J.Jr, “Global Positioning System: Theory and Applications”, Volume 1, AIAA, Washington DC, 1996.

[37] P. Maybeck, “Stochastic Models, Estimations and Control”, volume 1, 2. Academic Press, 1982.

[38] R. Brown and Y. Hwang, “Introduction to Random Signals and Applied Kalman Filtering”. Wiley, New York, 1992.

[39] B. D. Anderson and J. B. Moore, “Optimal Control”, Prentice Hall, Englewood Cliffs, NJ, 1990.

[40] K. Zhou, J. Doyle, and K. Glover. Robust and Optimll Control. Prentice Hall, Upper Saddle River, NJ, 1996.

[41] I. Horowitz, “Survey of quantitative feedback theory (QFT)”. International Journal of Control, 53(2):255-291, February 1991.

[42] M. A. Dahleh and J. B. Pearson, “Optimal feedback controllers for MIMO discrete-time systems”, IEEE Trans on Auto Control, April 1987.

[43] J. F. Shamma and M. Athans, “Analysis of gain scheduled control for nonlinear plants”, IEEE Trans. On Auto. Control, 35(8):898-907, 1990.

[44] W. J. Rugh, “Analytical framework for gain scheduling”. IEEE Control Systems Magazine, 11(1):79-84, 1991.

[45] P. Apkarian and P. Gabinet, “A convex characterisation of gain-scheduled H, controllers”. IEEE Trans. on Auto. Control, 40(5):853-864, May 1995.

[46] J-J. E. Slotine and W. Li, “Applied Nonlinear Control”, Prentice Hall, Englewood Cliffs, NJ, 1991.

[47] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, “Nonlinear and Adaptive Control Design”, John Wiley and Sons, New York, NY, 1995.

[48] K. S. Narendra. Parameter, “Adaptive control-the end ... or the beginning?”. In Proceedings of the 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, December 1994.

[49] J. R. Cloutier, C. N. D’Souza, and C. P. Mracek, “Nonlinear regulation and nonlinear H, control via the state-dependent Riccati equation technique; part 1, theory; part 2, examples”. In Proceedings of the International Conference on Nonlinear Problems in Aviation and Aerospace. Available through University Press, Embry-Riddle Aeronautical University, Daytona Beach, FL, 32114, May 1996.

[50] C. P. Mracek, J. R. Cloutier, and C. N. D’Souza. "A new technique for nonlinear estimation". In Proceedings of the IEEE Conference on Control Applications, Dearborn, MI, September 1996.

[51] J. R. Cloutier, "Adaptive matched augmented proportional navigation" Presented at the A IAA Missile Sciences Conference, November 1994.

[52] J. R. Cloutier "Time-to-go-less guidance with cross channel couplings" In Proceedings of the AIAA Missile Sciences Conference, Monterey, CA, December 1996.

[53] C. P. Mracek and J. R. Cloutier "Missile longitudinal autopilot design using the state-dependent Riccati equation method" In Proceedings of the International Conference on Nonlinear Problems in Aviation and Aerospace. Available through University Press, Embry-Riddle Aeronautical University, Daytona Beach, FL, 32114, May 1996.

[54] K. A. Wise, J. L. Sedwick, and R. L. Eberhardt "Nonlinear control of missiles" McDonnell Douglas Aerospace Report MDC 93B0484, October 1993.

[55] D. E. Williams, B. F'riedland, and A. N. Madiwale "Modern control theory for design of autopilots for bankto-turn missiles" Journal of Guidance, Control, and Dynamics, 10(4):378-386, July-August 1987.

[56] C. P. Mracek and J. R. Cloutier "A preliminary control design for the nonlinear benchmark problem" In Proceedings of the IEEE Conference on Control Applications, Dearborn, MI, September 1996.

[57] C. P. Mracek and J. R. Cloutier "Control designs for the nonlinear benchmark problem via the statedependent Riccati equation method" Submitted to the International Journal of Robust and Nonlinear Control.

[58] D. Ehrler and S. R. Vadali "Examination of the optimal nonlinear regulator problem" In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, August 1988.

[59] Vikas Kumar N "Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering" Aerospace Engineering, Department of Aerospace Engineering Indian Institute of Technology, BOMBAY Mumbai July 2004.

[60] James R.Cloutier "Navigation and Control Branch U.S. Air Force Armament Directorate" Eglin AFB, FL 32542-6810, Proceedings of the American Control Conference Albuquerque, New Mexico June 1997.

[61] James R.Cloutier, Curtis P.Mracek, D Brett Ridgely, Kelly D.Hammett "State-Dependent Riccati Equation Techniques: Theory and Applications" American Control Conference Workshop 1998.

[62] U Shaked and Y Theodor "$H\infty$-optimal estimation: a tutorial" IEEE Conference on Decision and Control, pp.2278-2286 (December 1992).

[63] M. Green and D. Limebeer "Robust Linear Control" Prentic-Hall, Englewood Cliffs, N.J, 1993.

[64] D. Limebeer, M. Green and D. Walker, "Discrete-time H∞ Control" Proc. 28th Conf. Decision and Control, pp. 292-296, 1989.

[65] U. Shaked and N. Berman "H∞ Nonlinear Filtering of Discrete-time Process" IEEE Trans. Signal Processing, vol. 43, pp. 2205-2209, 1995.

[66] I. Petersen and A. Savkin "Robust Kalman Filtering for Signals and Systems with Large Uncertainteis" Birkhauser, Boston, 1999.

[67] T. Basar and P. Baernard "H∞ Optimal Control and Related Minimax Design Problems" A Dynamic Game Approach. Systems and Control: Foundations and Applications. Birkhauser, 1991.

[68] G. Einicke and L. White "Robust Extended Kalman Filtering" IEEE Transactions on Signal Processing, Vol. 47, No. 9, pp. 2596 - 2599, 1999.

[69] Giorgio Grisetti "Simultaneous Localisation and Mapping" Towards a PhD Thesis, University of Rome "La Sapienza" in the Intelligent Systems Lab, April 2006.

[70] R. Smith, M. Self, and P. Cheeseman “A Stochastic Map for Uncertain Spatial Relationships” Proc. Fourth Int'l Symp. Robotics Research, 1987.

[71] Andrew J. Davison and David W. Murray, Member, IEEE “Simultaneous localisation and Map-Building Using Active Vision” IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 7, July 2002.

[72] C.G. Harris and J.M. Pike “3D Positional Integration from Image Sequences” Proc. Third Alvey Vision Conf., pp. 233±236, 1987.

[73] N. Ayache "Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception" Cambridge, Mass: MIT Press, 1991.

[74] H. F. Durrant-Whyte “Where am I? A Tutorial on Mobile Vehicle Localisation” Industrial Robot, vol. 21, no. 2, pp. 11±16, 1994.

[75] C.G. Harris, "Geometry from Visual Motion", Active Vision, pp 263 – 284 ISBN: 0-262-02351-2, 1993.

[76] P.A. Beardsley, I.D. Reid, A. Zisserman, and D.W. Murray “Active Visual Navigation Using Non-Metric Structure” Proc. Fifth Int'l Conf. Computer Vision, pp. 58±65, 1995.

[77] J.-Y. Bouget and P. Perona “Visual Navigation Using a Single Camera” ICCV5, pp. 645±652, 1995.

[78] C. Schmid, R. Mohr, and C. Bauckhage "Evaluation of interest point detectors" International Journal of Computer Vision, 37(2):151–172, June 2000.

[79] H. Moravec "Obstacle avoidance and navigation in the real world by a seeing robot rover" Technical Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute, 1980.

[80] Mikolajczyk and Schmid "An affine invariant interest point detector" In European Conference on Computer Vision (ECCV), Copenhagen, Denmark, pp. 128-142, 2002.

[81] David lowe “Distinctive Image Features from Scale-Invariant Keypoints” Computer Science Department University of British Columbia Vancouver, B.C, Canada lowe@cs.ubc.ca.

[82] Dan Simon “Optimal state estimation, Kalman, $H\infty$, and Nonlinear approaches” Welley interscience, A John wiley & sons, INC, Publication.

[83] LIU Qing , LIN Tu-sheng “The Corner Detection Algorithm Based on 2-D Discrete Wavelet Transform” The 3rd Intetnational Conference on Innovative Computing Information and Control (ICICIC'08), IEEE 2008.

[84] 9. J. Ferruz and A. Ollero "Real-time feature matching in image sequences for non-structured environments" applications to vehicle guidance. Journal of Intelligent and Robotic Systems, 28:85–123, 2000.

[85] David Capel “Image Mosaicing and Super-resolution” Edition: illustrated. Published by Springer ISBN 1852337710, 9781852337711, 2004.

[86] Deepesh Jain “EE392J Digital Video Processing” Stanford University, Stanford, CA, Mars 2004.

[87] P.J.S.G. Ferreira "Teaching signal and image reconstruction algorithms" Revista Do Detua, Vol. 1, N° 6, pp. 567-571, September, 1996.

[88] A. Papoulis "A New Algorithm in Spectral Analysis and Band-Limited Extrapolation" IEEE Transactions on Circuits and Systems 22(9), pp. 735-742, 1975.

[89] R.W. Gerchberg "Super-resolution through error energy reduction" Optica Acta 21(9), pp. 709-720, 1974.

[90] M. W. M. G. Dissanayake, P. Newman, S. Clark, and H. F. Durrant-Whyte “A solution to the simultaneous localisation and map building (SLAM) problem” IEEE Tran. Robot. Automat, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[91] P. Moutarlier and R. Chatila “Stochastic multisensory data fusion for mobile robot location and environment modeling” in Proc. 5th Int Symp. Robot. Research, Tokyo, Japan, 1989.

[92] T. Bailey, “Mobile robot localisation and mapping in extensive outdoor environments” Ph.D. dissertation, Univ. Sydney, Sydney, NSW, Australia, 2002.

[93] A. J. Davison and D. W. Murray “Simultaneous localisation and map building using active vision” IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):865-880, 2002.

[94] A. J. Davison “Real-Time Simultaneous Localisation and Mapping with a Single Camera” Proceedings of the 9th International Conference on Computer Vision, pp:1403-1410, 2003.

[95] T. Nir and A. M. Bruckstein "Causal Camera Motion Estimation by Condensation and Robust Statistics Distance Measures" Proceedings of the 8th European Conference on Computer Vision, 2004.

[96] Mikolajczyk, K., Schmid, C.A "performance evaluation of local descriptors" PAMI 27 (2005) 1615–1630.

[97] H. Bay, , A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008.

[98] Montemerlo, M, Thrun, S, Koller, D, Wegbreit, B "FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localisation and mapping that provably converges" In: Proc. 18th Int. Joint Conf. Artif. Intell. Acapulco, Mexico (2003).

[99]. Grisetti, G., Stachniss, C., Burgard, W "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling" In: Proc. IEEE Int. Conf. Robot. Automat, pp. 2443–2448. Barcelona, Spain (2005).

[100] A. Nemra and N. Aouf "Robust Airborne 3D Visual Simultaneous Localisation and Mapping with Observability and Consistency Analysis" J Intell Robot Syst, DOI 10.1007/s10846-008-9306-6.

[101] S. J. Kim "Efficient simultaneous localisation and mapping algorithms using submap networks" Ph.D. Dissertation, Dept. Ocean. Eng., Mass. Inst. Technol, Cambridge, May, 2004.

[102] S. J. Julier and J. K. Uhlmann "A counter example for the theory of simultaneous localisation and map building" in Proc. IEEE Conf. Robot. Automat, Seoul, Korea, May 21–26, 2001, pp. 4238–4243.

[103] J. A. Castellanos, J. Neira, and J. D. Tardos "Limits to the consistency of EKF-based SLAM" in 5th IFAC Symp. Intell. Autonom. Veh., IAV'04, Lisbon, Portugal, Jul. 2004.

[104] S. Huang, and G. Dissanayake "Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM" IEEE Transactions on Robotics, vol 23, No 5, 2007.

[105] M.L. Benmessaoud, A. Lamrani, K.Nemra and A.K.Souici "Single-Camera EKF-vSLAM" proceedings of world academy of science, engineering and technology PWASET Volume, ISSN 1307-6884, 30 July 2008.

[106] Joan Solà Ortega "Towards Visual Localisation, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach" PhD Dissertation, National Polytechnic Institute of Toulouse (INPT), February 2007.

[107] R. Smith, M. Self, and P. Cheeseman "Estimating uncertain spatial relationships in robotics" in Autonomous Robot Vehicles, I.J. Cox and G.T. Wilfon, Eds. New York: Springer-Verlag, pp. 167–193, 1990.

[108] M.Bryson and S.Sukkarieh "Observability analysis and Active control for airborne SLAM", IEEE Transactions on Aerospace and Electronic Systems 44(1) 2008.

[109] T.Vidal-Calleja, M.Bryson, S.Sukkarieh, A.Sanfeliu and J.Andrade-Cetto "On the Observability of Bearing-only SLAM" IEEE Conference on Robotics and Automation, 2007.

[110] J.A Castellanos, R.Martinez-Cantin, J.D. Tard`os and J.Neira “Robocentric map joining: Improving the consistency of EKF-SLAM” Robotics and Autonomous Systems, volume 55, pages 21–29, 2007.

[111] T.Bailey, J.Nieto, J.Guivant, M.Stevens and E.Nebot “Consistency of the EKF-SLAM Algorithm” IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.

[112] H .Singh, J. Catipovic, R. Eastwood, L. Freitag, H. Henriksen, F. Hover, D. Yoerger, 3 .G. Bellingham, and B.A. Moran “An integrated approach to multiple AUV communications, navigation and docking” In Proc. IEEE Oceanic Engineering Society, pp 59-64, 1996.

[113] J.H. Kim, J.M. DeFilipps, N.P. Impert, C.F. Derheim, M.Y. Thompson, S. Ray, and R.C. Butler "Atm network based integrated battle space simulation with multiple avawacs-fighter platforms" In Proc. IEEE Military Communications Conference, volume 1, pp 101-107. IEEE, 1998.

[114] E.W. Nettleton, H.F. Durrant-Whyte, P.W. Gibbens, and A.H. Goktogan. “Multiple platform localisation and map building”. In Proc. of Sensor Fusion and Decentralised Control in Robotic Systems III November, volume 4196, pp 337-347, 2000.

[115] W. Cohen “Adaptive mapping and navigation by teams of simple robots” Robotics and Autonomous Systems, vol. 18, pp. 411–434, 1996.

[116] K. Sty “Using situated communication in distributed autonomous mobile robots” in Seventh Scandinavian Conferenceon Arti£cial Intelligence (SCAI01), 2001.

[117] A.I. Mourikis, S.I. Roumeliotis "Performance Analysis of Multirobot Cooperative Localisation" IEEE Transactions on Robotics 22(4), pp. 666-681, Aug. 2006.

[118] S.I. Roumeliotis, I.M. Rekleitis "Propagation of Uncertainty in Cooperative Multirobot Localisation: Analysis and Experimental Results" Autonomous Robots**,** 17(1), pp. 41-54, July 2004.

[119] W. Burgard, D. Fox, S. Thrun "Active mobile robot localisation by entropy minimisation" Proceedings of Second Euro-micro Workshop on Advanced Mobile Robots (EUROBOT'97), 1997.

[120] Rui Rochaa,b, Jorge Diasa, Adriano Carvalhob "Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theory Robotics and Autonomous Systems" Pages 282-311, Vol:53, Issues 3-4, December 2005.

[121] Ming Wu, Feifei Huang, Long Wang and Jiyin Sun "Cooperative Multi-Robot Monocular-SLAM using Salient Landmarks" International Asia Conference on Informatics in Control, Automation and Robotics, IEEE 2009, pp151-155.

[122] Arturo Gil, Óscar Reinoso, Mónica Ballesta, Miguel Juliá "Multi-robot visual SLAM using a Rao-Blackwellized particle filter" Robotics and Autonomous Systems, Robotics and Autonomous Systems, Pages 68-80, Vol:58, Issues 1, January 2010.

[123] Martinelli, F. Pont, and R. Siegwart "Multi-robot localisation using relative observations" IEEE International Conference on Robotics and Automation, 2005.

[124] N. Trawny and T. Barfoot "Optimized motion strategies for cooperative localisation of mobile robots" In Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, pp. 1027–1032, 2004.

[125] Wu, Ming and Huang, Feifei and Wang, Long and Sun, Jiyin "Cooperative Multi-Robot, Monocular-SLAM Using Salient Landmarks" In Proceedings of the International Asia Conference on Informatics in Control, Automation and Robotics, pp: 151-155, 2009.

[126] N. Ergin Özkucur and H. Levent Akın "Cooperative Multi-robot Map Merging Using Fast-SLAM" Robot Soccer World Cup XIII pp:449-460 Volume 5949/2010 Lecture Notes in Computer Science Springer Berlin / Heidelberg, 2010-07-18, RoboCup 2009.

[127] M. W. M. G. Dissanayake, P. Newman, S. Clark, and H. F. Durrant-Whyte “A solution to the simultaneous localisation and map building (SLAM) problem” IEEE Tran. Robot. Automat, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[128] A. Nemra and N. Aouf "Robust Airborne 3D Visual Simultaneous Localisation and Mapping with Observability and Consistency Analysis" J Intell Robot Syst, DOI 10.1007/s10846-008-9306-6, June 2009.

[129] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “A probabilistic approach to collaborative multi-robot localisation,” Autonomous Robots, vol. 8, p.325344, 2000.

[130] S. Thrun, A probabilistic online mapping algorithm for teams of mobile robots, Int. J. Rob. Res. 20 (5), pp 335–363, 2001.

[131] Dellaert, F., and Kaess, M. Square Root SAM "Simultaneous Localisation and Mapping via Square Root Information Smoothing" The International Journal of Robotics Research, 2006, vol. 25, no. 12, pp.1181-1203.

[132] Kaess, M, Ranganathan, A, and Dellaert, F. iSAM "Incremental Smoothing and Mapping" IEEE Trans. On Robotics, Dec. 2008, vol. 24, no. 6, pp. 1365-1378.

[133] Andersson, L.A.A. Nygards, J C-SAM "Multi-Robot SLAM using square root information smoothing IEEE International Conference on Robotics and Automation", pp: 2798 – 2805, Pasadena, CA, USA, 2008.

[134] M. K. Pitt and N. Shephard "Filtering via simulation: auxiliary particle filters" J. Amer. Statist. Assoc, vol. 94, no. 446, pp. 590–599, 1999.

[135] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp "A tutorial on particle filter for on-line nonlinear/nongaussian bayesian tracking" IEEE Trans. Signal Process, vol. 50, no. 2, pp. 174–188, 2002.

[136] S. Julier and J. Uhlmann "A new extension of the Kalman filter to nonlinear systems" in Int Symp, Aerospace/Defense Sensing, Simul and Controls, Orlando, Proceedings Vol. 3068, pp.182-193, FL, 1997.

[137] L. Parker, "ALLIANCE: architecture for fault-tolerant multirobot cooperation" IEEE Trans. Rob. Autom. 14 (2) (1998) 220–240.

[138] D. Jung "An architecture for cooperation among autonomous agents" Ph.D. Thesis, Department of Computer Science, University of Wollongong, Australia, 1998.

[139] M. Matari´c, G. Sukhatme "Task-allocation and coordination of multiple robots to planetary exploration" Proceedings of 10th International Conference on Advanced Robotics, 2001.

[140] B. Gerkey, M. Matari´c, Sold "Auction methods for multirobot coordination" IEEE Trans. Rob. Autom. 18 (October) (2002) 758–768.

[141] R. Brooks "A robust layered control system for a mobile robot" IEEE J. Rob. Autom, RA-2 (1986) 14–23.

[142] R. Arkin "Motor schema based mobile robot navigation" Int. J.Rob. Res. 8 (4) (1989) 92–112.

[143] Y. Cao, A. Fukunaga, A. Kahng "Cooperative mobile robotics: antecedents and directions" Auton. Robots 4 (1997) 1–23.

[144] Uwe D. Hanebeck, Kai Briechle, and Joachim Hom "A tight bound for the joint covariance of two random variables with unknown but constrained cross-correlation" In Proc. IEEE Int. Conf: Multisensor Fusion Integr Intell. Syst. (MFI '01), pages 85-90, August 2001.

[145] Simon J. Julier and Jeffrey K. Uhlmann “Generalized decentralised data fusion with covariance intersection” In David L. Hall and James Llinas, editors, Handbook of Multisensor Data Fusion, pages 12-1-12-25, CRC Press, London, New York, Washington, D. C, 2001.

[146] Chee-Yee Chong and Shozo Mori “Convex combination and covariance intersection algorithms in distributed fusion” In Proc. Int. Conf: Inf: Fusion (FUSION '01), August 2001.

[147] Xiao-Rong Li and P. Zhang. “Optimal linear estimation fusion-part III: Cross-correlation of local estimation errors” In Proc. Int. Conf: Inf: Fusion (FUSION '01), pages WeB 1-1 1-WeB 1-18, August 2001.

[148] Wolfgang Niehsen “Information fusion based on fast covariance intersection filtering” In Proc. Int. Conf.Inf. Fusion (FUSION '02), pages 901-905, July 2002.

[149] Dietrich Franken, Andreas Hupper “Improved Fast Covariance Intersection for Distributed Data Fusion” In IEEE proc, 7th International Conference on Information Fusion (FUSION), 2005.

[150] M Vishwanath, "The recursive pyramide algorithm for the discrete wavelet transform" IEEE Transaction, Signal processing, Vol 42, n°3, pp 673-676, Mars 1994.

[151] Mallat, S.G "Multifrequency Channel Decompositions of Images and Wavelet Models" IEEE Trans. On ASSP, 37 (1989) 2091-2110.

[152] Irie, K., Kishimoto, R "A Study on Perfect Reconstructive Sub-band Coding" IEEE Trans.on CAS for Video Technology, 1 (1991) 42-48.

[153] http://www.robots.ox.ac.uk/~vgg/research/affine/