DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
ESCUELA SUPERIOR DE INGENIEROS
UNIVERSIDAD DE SEVILLA

Técnicas basadas en homografía para la autolocalización de vehículos autónomos aéreos utilizando secuencias de imágenes monoculares

Homography based techniques for unmanned aerial vehicle self-localization using monocular imagery

por

Fernando Caballero Benítez

PROPUESTA DE TESIS DOCTORAL
PARA LA OBTENCIÓN DEL TÍTULO DE

DOCTOR INGENIERO DE TELECOMUNICACIÓN

SEVILLA, OCTUBRE 2007

Directores

Dr.-Ing. Aníbal Ollero Baturone, Catedrático de Universidad

Dr.-Ing. Joaquín Ferruz Melero, Profesor Titular de Universidad

UNIVERSIDAD DE SEVILLA

Memoria para optar al grado de Doctor Ingeniero de Telecomunicación por la Universidad de Sevilla

Autor:	**Fernando Caballero Benítez**
Título:	**Técnicas basadas en homografía para la autolocalización de vehículos autónomos aéreos utilizando secuencias de imágenes monoculares**
Departamento:	**Departamento de Ingeniería de Sistemas y Automática**

V° B° Director:

Aníbal Ollero Baturone

V° B° Director:

Joaquín Ferruz Melero

El autor:

Fernando Caballero Benítez

A mis padres

A Vicky

Abstract

The aim of this Thesis is to develop computer vision-based techniques for localizing an unmanned aerial vehicle (UAV) by means of an on-board camera. Only natural landmarks provided by a feature tracking algorithm will be considered, without the help of visual beacons or landmarks with known positions.

The first technique proposed in this Thesis is a monocular visual odometer which could be used as a backup system when the accuracy of GPS is reduced to critical levels. The goal is the development of artificial vision techniques for the computation of the relative translation and rotation from the images gathered by a camera attached to the UAV. The analysis of the problem takes into account the stochastic nature of the estimation and practical implementation issues.

Homographies will be used as formal representation of motion between consecutive images when the imaged scene in the camera can be assumed to be planar. The experimental results presented in this Thesis show that this assumption holds when the UAV flies at a relatively high altitude, and proposes mechanisms to extend its validity to pseudo-planar scenes.

Approaches to reduce the impact of cumulative errors in position estimation are also proposed. The Thesis describes methods to build on-line mosaics and use them as a consistent description of the UAV environment. This representation helps to quantify and correct the drift in the position estimation by comparing the currently estimated position to the one which can be computed from the mosaic. In addition, a new stochastic framework is proposed for online mosaic building to extend these corrections to the whole mosaic; this mosaic representation allows to recompute the relative position of the images when the UAV visits previously registered areas.

The visual odometer is also integrated in a simultaneous localization and mapping (SLAM) scheme in order to provide localization capabilities when the mosaic building process cannot be carried out. Novel prediction and landmark initialization for SLAM in UAVs are presented.

The Thesis is supported by an extensive experimental work where the proposed algorithms have been tested and validated using real UAVs.

Acknowledgements

This document summarizes my work in the last four years. Although many efforts have been dedicated to this Thesis, they are nothing compared to the support provided by my family, friends and colleagues. Their help and good advices constitute the basis of my research. Thanks to all of them.

I would like to also thank the collaboration of the Laboratoire d'Analyse et d'Architecture des Systèmes in Toulouse (Francia), and specially Simon Lacroix. The images gathered by their Blimp, KARMA, made possible some of the experimental results presented in this Thesis.

I want to emphasize and thank the great work carried out by my advisors Aníbal Ollero and Joaquín Ferruz, and of course the rest of the Robotics, Visión and Control Research Team at the University of Seville, specially Luis, Ivan, Manolo and Angel.

Finally, this Thesis has been widely supported by national and international funded research projects related to UAVs. It is worth to mention the valuable support of the COMETS (IST-2001-34304) and AWARE (IST-2006-33579) projects funded by the European Commission, the AEROSENS (DPI-2005-02293) project funded by the Spanish Government, and the SADCOM (TEP-2005-375) project funded by Junta de Andalucía (Spain).

Contents

List of Figures

Chapter 1

Introduction

This Chapter introduces the Thesis motivation, objectives and scope. Current work on visual based localization are described and linked to the proposed approach.

1.1 Motivation

Outdoor robotics applications in natural environments usually require higher mobility than the motion capabilities provided by existing ground robotic vehicles. In fact, in spite of the progress in the development of unmanned ground vehicles along the last 20 years, navigating in unstructured natural environments still poses significant challenges. The existing ground vehicles have inherent limitations to reach the desired locations in many applications. The characteristics of the terrain and the presence of obstacles, together with the requirement of fast response, may represent a major drawback to the use of any ground locomotion system. Thus, in many cases, the use of aerial vehicles is the only effective way to reach the target to get information or to deploy instrumentation.

Unmanned aerial vehicles (UAVs) are suitable for many robotic systems because they are not affected by the above mentioned limitations of ground vehicles.

In the last ten years UAVs have improved their autonomy both in energy and information processing. Significant achievements have been obtained in autonomous positioning and tracking. These improvements are based on modern satellite-based

position technologies, inertial navigation systems, communication and control technologies, and image processing. Furthermore, new sensing and processing capabilities have been implemented on-board the UAVs. Thus, today we can consider some UAVs as intelligent robotic systems integrating perception, learning, real-time control, situation assessment, reasoning, decision-making and planning capabilities for evolving and operating in complex environments.

In most cases, UAVs use the Global Position System (GPS) to determine their position. As pointed out in the Volpe Report (Volpe, 2001), the accuracy of this estimation directly depends on the number of satellites used to compute the position and the quality of the signals received by the device; radio effects like multi-path propagation could cause the degradation in the estimation. In addition, radio frequency interferences with coexisting devices like cellular phone, television broadcasting or, in general, VHF communications, could make the position estimation unfeasible.

These problems are well known in robotics. Thus, odometry (estimation of the relative position with respect to a starting point by means of local sensors located in the wheels of the robot, usually by measuring velocity and orientation) is commonly used in terrestrial robots as a backup positioning system or in sensor data fusion approaches. This local estimation allows temporally managing GPS faults or degradations. However, the lack of odometry systems in most aerial vehicles can lead to catastrophic consequences under GPS errors; incoherent control actions could be commanded to the UAV, leading to crash and the loss of valuable hardware.

If small UAVs are considered, their low payload represents a hard restriction on the variety of devices to be used for odometry. Sensors like 3D or 2D laser scanners are too heavy and have an important dependence to the UAV distance to the ground. Although there exist small devices for depth sensing, their range is usually shorter than fifteen meters. Stereo vision systems have been successfully applied to low/medium size UAVs due to its low weight and versatility, but the rigid distance between the two cameras limits the useful altitude range.

Monocular vision seems to offer a good solution in terms of weight, accuracy and scalability. However, although several approaches have been presented, mainly based on the existence of visual beacons placed in known positions, the use of natural

landmarks for this purpose is very challenging and has been barely addressed by the research community.

This Thesis proposes a monocular visual odometer and vision-based localization methods to act as backup systems when the accuracy of GPS is reduced to critical levels. The objective is the development of computer vision techniques for the computation of the relative translation and rotation, and for the localization of the vehicle based on the images gathered by a camera on-board the UAV. The analysis of the problem takes into account the stochastic nature of the estimation and practical implementation issues.

1.2 Related work

First researches on vision applied to UAV position estimation starts in the nineties at the Carnegie-Mellon Univerisity (CMU). In (Amidi et al., 1998), it is described a visual based odometer that allowed to lock the UAV to ground objects and sense relative helicopter position and velocity in real time by means of stereo vision. The same visual tracking techniques, combined with inertial sensors, were applied to autonomous take off, following a prescribed trajectory and landing. The CMU autonomous helicopter also demonstrated autonomous tracking capabilities of moving objects by using only on-board specialized hardware.

Stereo vision has also been used in the GTMax helicopter at Georgia Tech (Johnson and Schrage, 2003) where the VISTA system (VISual Thread Awareness) is designed for real time obstacle detection and avoidance using a small commercial stereo vision device. The approach, detailed in (Byrne et al., 2006), combines the existing real time stereo system with a global segmentation step to increase robustness to matching errors, resulting in a practical obstacle detection system.

Vision systems have been actively used for autonomous landing in targets. The BEAR project at Berkeley is a good example. In this project vision-based pose estimation of unmanned helicopters relative to a planar landing target and vision-based landing of an aerial vehicle on a moving deck have been researched (Shakernia et al., 2002; Vidal et al., 2002). A technique based on multiple view geometry of the

same plane, in this case the planar target, is used to compute the real motion of the UAV with respect to the target.

Computer vision is also used for safe landing in the AVATAR project. Thus, in (Garcia-Pardo et al., 2001), a strategy and an algorithm relying on image processing techniques to search the ground for a safe landing spot is presented. Vision-based techniques for landing on a helipad of known shape are presented in (Saripalli et al., 2003; Saripalli and Sukhatme, 2003), where the case of landing on a slow moving helipad is also considered.

Simultaneous Localization and Mapping (SLAM) techniques are closely related to the position estimation problem. Although vision-based SLAM has been widely used in ground robots and has demonstrated its feasibility for consistent perception of the environment and position of the robot, only a few applications have been implemented on UAVs. The researches carried out in the LAAS laboratory in France and the Centre for Autonomous Systems in Australia can be highlighted. The first of them has developed an stereo vision system designed for the KARMA blimp (Lacroix et al., 2002; Hygounenc et al., 2004) where interest point matching and Kalman filtering techniques are used for simultaneous localization and mapping with very good results. However, this approach is not suitable for helicopters, as the baseline of the stereo rig that can be carried is small, and therefore it limits the height at which the UAV can fly. UAV simultaneous localisation and map building with vision using a delta fixed wing platform is also presented in (Kim and Sukkarieh, 2004) at the Centre for Autonomous Systems. Artificial landmarks of known size are used in order to simplify the landmark identification problem. The known size of the landmarks allows to use the cameras as a passive range/bearing/elevation sensor.

The use of an omnidirectional camera for helicopter control has been considered in (Hrabar and Sukhatme, 2003). The camera is used to maintain the helicopter in the centroid of a set of artificial targets. The processed images are directly used to command the helicopter. The paper shows the feasibility of the procedure, but no actual control is tested. Omnidirectional vision is also used in (Demonceaux et al., 2006) to estimate the attitude of an UAV. The method detects the horizon line by means of image processing and computes the attitude from its apparent motion.

Although the technique seems to be reliable, the authors do not compare the results with any ground truth, and the accuracy of the method cannot be measured.

In the work of (Mejías et al., 2006), vision is used to track features of buildings. Image features and GPS measurements are combined together to keep the UAV aligned with the selected features. Although visual servoing has been widely used in indoor robotics, as in (Pari et al., 2006) where it is applied to the tracking of a 3DOF joint system, the application to aerial robotics is still scarce. Thus, Control design and stability analysis of image-based controllers for aerial robots are presented in (Mahony and Hamel, 2005), while (Proctor et al., 2006) presents recent work on vision-based control of a fixed wing aircraft.

Finally, localization techniques based on monocular imagery and online mosaicking have been proposed for AUVs (autonomous underwater vehicles) in (Garcia et al., 2001) and (Garcia et al., 2002). Although the environment is clearly different, the limitations are similar. The method uses an online built mosaic as environment representation and the relations between images to compute the translation of the robot. However, only the position of the images are considered, and the motion computation problem is barely described.

1.3 Thesis scope and layout

The aim of the Thesis is to develop computer vision-based techniques for localizing an UAV by means of an attached camera. Only the natural landmarks provided by a feature tracking algorithm will be considered, so visual beacons or landmarks with known positions are discarded. The algorithms will be developed in the framework of the Projective Geometry which has demonstrated to be a powerful tool for representing the image projection geometry and many others problems like stereo vision (Faugeras and Luong, 2001), image back-projection, multiple view geometry (Hartley and Zisserman, 2004) or camera calibration (Zhang, 1999).

In this sense, the Thesis firstly addresses in Chapter 2 an introduction to Projective Geometry and, more specifically, to Homography computation. Homography will be used as formal representation of motion between consecutive images when the imaged

scene in the camera can be assumed to lay in a plane. The experimental results presented in this document will show how this assumption holds when the UAV flies at relatively high altitude. The algorithm for Homography computation has been tested and validated with thousands of images captured by different UAVs flying at different altitudes, from fifteen to one hundred and fifty meters. This algorithm was briefly described in (Ollero et al., 2004) and it is similar to the approach presented by Bosch in (Bosch et al., 2006). The technique has been successfully tested during the COMETS project (Ollero et al., 2005) for on-line compensation of the induced motion in the images by UAVs vibrations in (Merino et al., 2006a), and for fire detection and monitoring in (Martinez et al., 2006). The Chapter deals also with the computation of the covariance matrix associated to the estimated homography. This information is used all along the Thesis to propagate errors and to know how accurate the estimated homography is.

Chapter 3 details a method that allows to compute the real camera motion by means of the estimated homography; a homography based odometer. The Chapter extends the work presented in (Caballero et al., 2005), probably one of the first applications of Homography to the UAV motion estimation using natural landmarks. Although the theoretical background was introduced at the beginning of the eighties by Tsai (Tsai et al., 1982), the Chapter particularizes the method to UAVs and provides a study about the uncertainties associated to the estimated motion. Real images and telemetry gathered by an UAV are used to validate the algorithm. Experimental results show the presence of accumulative errors, typical in odometric approaches, that make the position estimation diverge through time.

Chapter 4 presents a new approach to reduce the impact of the accumulative errors in position estimation by means of online mosaicking. The Chapter describes the method to build a mosaic given the image-to-image homographies. The mosaic is used to create a consistent view of the environment of the UAV and hence, to detect the drift in position estimation by using the mosaic as a resource. This technique, presented in (Caballero et al., 2006) and used together with the position estimation in (Merino et al., 2006b), is specially suitable for monitoring and surveillance tasks in

which the UAV will repeatedly cover the same area. The Chapter shows experimental results where the benefits with respect to the odometric approach are evident.

However, the static nature of the mosaic built with the approach of Chapter 4 makes impossible its improvement once it is built, even when new accurate information is available. This issue is addressed in Chapter 5, and a new stochastic framework for mosaic building is proposed. The Chapter shows how the covariance matrix of the Homography can be used to establish stochastic relations among the images that compose the mosaic in such a way that when an area is revisited by the UAV, the mosaic can be updated by using the close-loop information. In addition, Chapter 5 proposes a new approach for homography computation based on a hierarchy of models that allows to compute the Homography when the UAV flies at low/medium altitude. Chapter 5 extends the work presented in (Caballero et al., 2007) and provides more experimental results that validate the algorithm.

Chapters 4 and 5 consider that the UAV flies in such a way that a mosaic can be computed and then used to refine the localization. The main constraints of this method are the associated to the mosaic building process. In general, the accuracy of the mosaic-based localization technique is small when the camera on the UAV does not point perpendicularly to the scene. In addition, the UAV must fly at approximately the same altitude in order to guarantee the mosaic coherence. Chapter 6 extends the localization to a more general problem by applying SLAM (simultaneous localization and mapping) techniques in order to provide localization capabilities when the mosaic building process can be carried out.

Finally, Chapter 7 summarizes the contributions of the Thesis and describes future work and research activities related to the UAV localization by means of monocular vision.

1.4 Main contributions

This Thesis presents several contributions in the Field of unmanned aerial vehicles. It includes the implementation of an odometry-based system for UAVs using monocular imagery and natural landmarks. The method is robust and can be easily implemented

to provide real-time UAV position estimation. The experimental results presented in the Thesis show the feasibility of the approach and demonstrate that it can be used as an effective backup of GPS.

The homography-based motion estimation has many applications to UAVs, particularly to helicopters. Frequent manoeuvres like hovering, take-off and landing, could be safer if the autonomous system is aided with relative positioning based on local perception, and it does not only rely on GPS.

The second contribution of this Thesis is the development of a mosaic-based localization for UAVs. The method allows to reduce the existing accumulated drift in pure odometry-based localization algorithms. Similar approaches have been proposed for underwater vehicles, but the work presented in this Thesis significantly extends the state of the art by including full homography correction estimations, and a novel stochastic method for on-line mosaic building that allows taking into account improvements not only in the position estimation, but also in the mosaic building procedure. Although the proposed mosaic building approach is applied on-line, the technique can also be used off-line to build super-resolution mosaic images for environment representation, geographical information systems (GIS) or other commercial purposes.

Finally, some contributions to the SLAM (Simultaneous Localization And Mapping) problem are presented. The homography-based odometer is integrated into a SLAM scheme as the main prediction hypothesis. In addition, a novel landmark initialization, based on the use of the normal vector to the scene plane, is described and used to reduce the landmark uncertainty in depth. The complete approach allows to construct a full vision-based monocular SLAM for UAVs. The implementation of these techniques is scarce in the bibliography.

1.5 Thesis framework

This Thesis has been mainly developed in the Robotics, Vision and Control Research Team at the University of Seville, in the framework of several national and international research projects related with UAVs. The support of these projects was

fundamental, in the sense that they provided aerial robots where to test part of the algorithms presented in this Thesis. They can be summarized as follows:

- COMETS project (IST-2001-34304). Funded by the European Commission. This project is currently finished. The main objective was to design and implement a distributed control system for cooperative detection and monitoring using heterogeneous Unmanned Aerial Vehicles (UAVs). Particularly, two helicopters and an airship were included. In order to achieve this general objective, the project designed and implemented a new control architecture, developed new control techniques, and integrated distributed sensing techniques and real-time image processing capabilities. COMETS demonstrated the proposed system and architecture in forest fire applications.

- AEROSENS project (DPI-2005-02293). Funded by the Spanish Government. The project aims at the development of a system based on the use of aerial and ground robots and sensor networks for cooperative perception. The system is based on the joint application of Aerial Robotics and the technology of Wireless Sensor Networks. Several autonomous and teleoperated vehicles are used including an helicopter and an airplane developed from the adaptation for autonomous operation of a radio controlled comercial airplane. The helicopter manoeuvrability and hovering capability allow reaching locations with constrained accessibility, collecting information and communicating with the on-ground sensors.

- SADCOM project (TEP-2005-375). Funded by Junta de Andalucía (Spain). This project aims at the development of distributed and autonomous systems technologies for the preservation of natural environments. The application of wireless sensor networks and autonomous vehicle technologies for wildlife monitoring and forest fire fighting. Thus, the project considers the use of wireless sensor networks, ground robots and aerial robots for sensing environmental variables.

- AWARE project (IST-2006-33579). Funded by the European Commission. The general objective of the project is the design, development and experimentation of a platform providing the middleware and the functionalities required for the cooperation among autonomous helicopters and a ground sensor-actuator wireless network, including mobile nodes carried by people and vehicles. The platform enables the operation in sites with difficult access and without communication infrastructure. Then, the project considers the self-deploying of the network by means of autonomous helicopters with the ability to transport and deploy loads (communication equipment and nodes of the ground network).

In addition, the Thesis was partially developed in the Laboratoire d'Analyse et d'Architecture des Systèmes in Toulouse (Francia) during a six months stay. Part of the stochastic representations showed in the Thesis were developed during this stay.

Chapter 2

Robust homography estimation

As stated in Chapter 1, in this thesis the homography will be used as a basic tool for estimating the motion that an UAV undergoes by using monocular image sequences. Although the homography model only holds when the imaged scene is planar or pseudo-planar, a very common situation when the UAV flies at high altitude, this Chapter will propose methods to extend the applicability of the homography to non-planar scenes in order to be able to perform motion estimation at medium or even low UAV altitude.

The homography will model the motion of a set of interest points between two images. This feature tracking is out of the scope of this thesis; and the existence of a functionality able to extract and match image features will be assumed. Particularly, the algorithm and implementation described in (Ollero et al., 2004) will be used.

Thus, this Chapter describes an algorithm to estimate the homography that relates two images of the same pseudo-planar scene. The covariance matrix will be used as a metric for measuring the accuracy of the estimation.

Firstly, the projective geometry fundamentals will be described in order to easily understand the homography estimation process. Then, the homography and covariance matrix estimation are detailed.

2.1 Matrix representation and conventions

A set of conventions will be used for the matrix and vector representation all along the Thesis document. This conventions help to better understand the equations:

- Matrix will be represented by capital bold letters such as $\mathbf{M}$, $\mathbf{H}$ or $\mathbf{Q}$.
- Vector will be represented by lower case bold letters such as $\mathbf{v}$ or $\mathbf{x}$
- By default, any vector is a column vector.
- Any scalar variable will be represented by normal italic letters such as k or j.

2.2 Projective geometry overview

2.2.1 Definitions

A projective transformation can be defined as an invertible application of $\mathbf{P^2}$ in $\mathbf{P^2}$ that applies lines into lines. This transformation are usually called homography or projectivity.

Some basic properties of the homography are the following:

- Any homography can be represented as a linear and invertible transformation in homogeneous coordinates. Inversely, any transformation of this nature can be considered as a homography.
- A homography between two planes is a linear transformation between three-dimensional homogeneous vectors $\tilde{\mathbf{x}}$, represented by the 3×3 $\mathbf{H}$ matrix such as $\tilde{\mathbf{x}}' = \mathbf{H}\tilde{\mathbf{x}}$.
- Given the homogeneous nature of the homography $\mathbf{H}$, it can be multiplied by an arbitrary constant $k\neq0$ and represent the same transformation. This means that the matrix $\mathbf{H}$ is constrained by eight independent parameters and a scale factor.

Depending on the kind of transformation, the projective transformations have different properties. Here, a subset of important transformations are detailed.

2.2.2 Isometry

Isometries are $\mathbf{R^2}$ plane transformations that preserve the Euclidean distance. An isometry is represented by the following matrix:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \varepsilon\cos\theta & -\sin\theta & t_x \\ \varepsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

Where $\varepsilon = \pm 1$. If $\varepsilon = 1$ then the isometry preserves orientation and it is an Euclidean transformation. If $\varepsilon = -1$, the isometry will invert the plane orientation.

The most important isometries are the Euclidean transformations. Thus, they can be decomposed as:

$$\tilde{\mathbf{x}}' = \mathbf{H_E}\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0^T} & 1 \end{bmatrix} \tilde{\mathbf{x}} \tag{2.2}$$

Where $\mathbf{R}$ is a 2×2 rotation matrix (so $\mathbf{R^T R} = \mathbf{R R^T} = \mathbf{I}$), $\mathbf{t}$ is a translation vector composed by two elements and $\mathbf{0}$ is a null vector composed by two elements.

An Euclidean transformation between planes have three degrees of freedom, one for the rotation and two for the translation. Given that each match provides two constraints in the transformation, only two matches are needed to estimate the Euclidean transformation between two planes.

2.2.3 Similarity

A similarity is an Isometry joint to an isotropic scale. If the Isometry is an Euclidean transformation, the Similarity can be decomposed as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.3}$$

By rearranging the equation:

$$\tilde{\mathbf{x}}' = \mathbf{H_S}\tilde{\mathbf{x}} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0^T} & 1 \end{bmatrix} \tilde{\mathbf{x}} \tag{2.4}$$

Where s represents the scale factor.

Similarities preserve the shape of the transformed plane. They have four degrees of freedom and hence can be computed by using two matches between planes.

2.2.4 Affinity

An Affine transformation or Affinity is defined as a non-singular linear transformation followed by a translation. The matrix representation of the Affinity is the following:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.5}$$

By rearranging the equation:

$$\tilde{\mathbf{x}}' = \mathbf{H_A}\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0^T} & 1 \end{bmatrix} \tilde{\mathbf{x}} \tag{2.6}$$

Where $\mathbf{A}$ a 2×2 non-singular matrix.

An Affine transformation in the plane has six degrees of freedom due to the four elements of the matrix $\mathbf{A}$. Thus, computing the Affinity that relates two planes requires at least three matches. These two new degrees of freedom allow to introduce plane distortions that preserve parallelism (see Fig. 2.1)

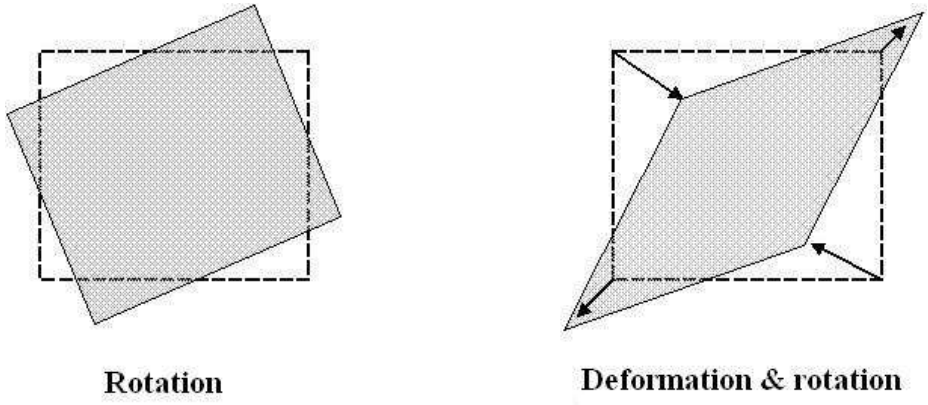


Figure 2.1: Typical Affine transformation

2.2.5 Projectivity

Projectivities generalize projective transformations. They can be expressed in the following matrix notation:

$$\tilde{\mathbf{x}}' = \mathbf{H_P}\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v^T} & v \end{bmatrix} \tilde{\mathbf{x}} \qquad \mathbf{v} = [v_1, v_2]^T \tag{2.7}$$

The matrix has nine elements but only eight are independent. Although it is usual to fix the scale factor as $v = 1$, it is not always correct because the scale factor could be zero for a valid homography. The Projectivity can be calculated if four matches are computed between the planes.

It can be seen how the main different between Projectivity and Affinity is the vector $\mathbf{v} = [v_1, v_2]^T$. In the Affinity this vector is fixed and equal to $[0, 0]^T$ but in the Projectivity is determines the non-linear behavior in the non-homogeneous space.

The more important invariant in projectivities is the cross relation among four aligned points that can be defined as:

$$RC(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}) = \frac{|\overline{\mathbf{x}}_\mathbf{1}\ \overline{\mathbf{x}}_\mathbf{2}||\overline{\mathbf{x}}_\mathbf{3}\ \overline{\mathbf{x}}_\mathbf{4}|}{|\overline{\mathbf{x}}_\mathbf{1}\ \overline{\mathbf{x}}_\mathbf{3}||\overline{\mathbf{x}}_\mathbf{2}\ \overline{\mathbf{x}}_\mathbf{4}|} \tag{2.8}$$

A Projectivity can be decomposed in several transformations (Isometry, Affinity and Projectivity respectively) as follows:

$$\mathbf{H} = \mathbf{H_S H_A H_P} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0^T} & 1 \end{bmatrix} \begin{bmatrix} s\mathbf{K} & \mathbf{t} \\ \mathbf{0^T} & 1 \end{bmatrix} \begin{bmatrix} s\mathbf{I} & \mathbf{t} \\ \mathbf{v^T} & v \end{bmatrix} = \begin{bmatrix} s\mathbf{A} & \mathbf{t} \\ \mathbf{v^T} & v \end{bmatrix} \tag{2.9}$$

Where $\mathbf{A}$ is a non-singular matrix defined by $\mathbf{A} = s\mathbf{RK} + \mathbf{tv^T}$ and $\mathbf{K}$ is an upper-triangular matrix normalized such as $det(\mathbf{K}) = 1$. This decomposition is valid if $v \neq 0$ and unique if s is positive.

2.3 Homography estimation

Given a set of matches between two images, this section will deal with the estimation of the homography that better fits their apparent motion. However, the features could experiment variations in theirs position caused not only by the camera motion. For instance, features associated to mobile objects or areas where the planar assumption does not hold.

Thus, the first step in the homography estimation will be to filter the set of matches in order to detect and eliminate erroneous matches. In this sense, algorithms like Least Median of Squares (LMedS) (Rousseeuw and Leroy, 1987) or Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981) have demonstrated very good results in model fitting. Both are randomized algorithms and have similar performance but, for the particular problem of homography fitting, LMedS has achieved slightly better results.

A M-Estimator (Zhang, 1995) will be applied over the set of filtered data to finally estimate the homography matrix. This second step allows precise estimation while detecting potentially erroneous or bad constrained matches at the same time.

2.3.1 Outlier rejection based on LMedS

Given a data set of n elements, the objective is to estimate the model that best fit to them. This model will be a projectivity or homography, that is, a 3×3 matrix defined

up to a scale factor. For simplicity and without loss of generality, the scale factor will be fixed such us $h_{33} = 1$, then:

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{2.10}$$

Given a data set ξ composed by n elements and a model to fit ζ, any element of ξ that does not fit the proposed model ζ will be named outlier. In the case that ξ is composed by n matches between two images and the model is the homography, outliers can be identified as:

- The match is erroneous, it means, the tracking algorithm makes an error and generates a false match. This kind of matches are really important because may increase the mean error of the fitted model if they are included in the estimation

- The match is correct but it doesn't fit the model. The homography assumes that the motion between two images can be modeled as a plane to plane transformation. This assumption will not hold when the scene is not planar and when there are mobile objects. These matches might introduce serious distortions in the estimated model, so it is important to remove them from the data set.

The Least Median of Squares method estimates the model by solving the following non-linear minimization problem:

$$\min \underset{i}{\text{med}}(r_i^2) \tag{2.11}$$

Thus, it will compute the minimum median of the squared residues applied to the whole data set. The value of the minimum median can be computed by searching the best homography estimation in the data set, however the computational cost of evaluating all possible data combinations could be extremely high. In this cases, randomized methods usually provide a good compromise between accuracy and computational cost.

In (Zhang, 1995) a randomized method is proposed to solve the minimization process. Given a set of n matches $\mathbf{c}_i$ and taking into account (2.11), the LMedS algorithm can be described as follows:

1. First, a Montecarlo-like algorithm is used to randomly select m subsets of p elements, where m will depend on the outlier detection likelihood and p is the smallest subset to compute the model.

2. For each of the m subsets, indexed by j, the best fit homography matrix $\mathbf{H}_j$ is computed

3. For each $\mathbf{H}_j$, the median of the squared residues, M_j, with respect the matches is computed.

$$M_j = \underset{i=1\ldots n}{\text{med}} \left(r_i^2(\mathbf{H}_j, \mathbf{c}_i)\right) \tag{2.12}$$

   Several techniques can be used to compute the value of r_i. In this case, the Euclidean distance between the computed position with the homography $\mathbf{H}_j$ and the match $\mathbf{c}_i$ will be used.

4. Finally, the smallest M_j is chosen from the data set. This minimum will be called M

$$M = \min_{j=1\ldots m} M_j \tag{2.13}$$

The number of subsets, m, has to be carefully determined in order to ensure low computation load and good model estimation. Thus, assuming that there exists a percentage E of outliers in the set of matches, the likelihood of having at least one of the m subsets without outliers is:

$$Pb = 1 - (1 - (1 - E)^p)^m \tag{2.14}$$

The value of Pb must be close to one, so:

$$m = \frac{1 - Pb}{1 - (1 - E)^p} \tag{2.15}$$

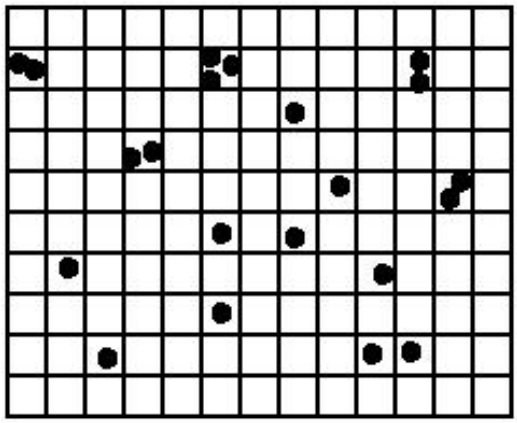


Figure 2.2: Bucketing example

Then, this equation allows to compute the number m of subsets to detect a percentage E of outliers with probability Pb in a data set of n elements.

As pointed out in (Rousseeuw and Leroy, 1987), LMedS has a poor efficiency in presence of Gaussian noise. The efficiency of a method is defined as the ratio between the lowest achievable variance for the estimated parameters and the actual variance provided by the given method. To compensate for this deficiency, a weighted least squares procedure is carried out. The robust standard deviation estimate is given by:

$$\sigma = 1.4826 \left(1 + \frac{5}{n-p}\right) \sqrt{M} \tag{2.16}$$

The value of σ can be used to define a threshold: every datum for which the residue r_i^2 is greater than $2.5\sigma^2$ is considered as outlier, and a right datum otherwise.

LMedS usually presents problems due to local minima fitting when the data set is sparse. To avoid this, in (Zhang, 1995) a bucketing approach is used to spatially divide the data set as shown in Fig. 2.2. Each bucket is associated to the set of data that falls into and empty buckets are excluded. Finally, p buckets are randomly selected; each of them will provide one of the p matches, randomly chosen among those associated to it. It is clear that the number of buckets is a key issue that depends on the model to be fitted

Implementation

The implementation is slightly different than the above algorithm due to the inclusion of some heuristic rules in the outlier rejection policy. Thus, the LMedS algorithm won't be applied to the overall data set, but to a specific subset. The improvement consists of remembering through time the outlier condition, that is, if a match was considered as an outlier in the last homography computation then it will probably be an outlier in the present computation. In addition, new matches won't be considered in LMedS because there is no available information to know if the match is reliable.

Therefore, matches will be classified in three possible categories:

- Erroneous matches. The matches that the feature tracking system marks as erroneous.
- Suitable matches for LMedS. They are features correctly matched that were not considered as outliers in previous computations.
- Suitable matches for M-Estimator. These matches are correct but they were outliers in the previous computation, or they are new matches.

The classification is depicted in Fig. 2.3, where the whole homography computation processing is summarized. It can be seen how some matches are used for the computation of LMedS, others are directly included in the outlier rejection process and the rest are classified as non-valid matches.

The model to be fitted in the LMedS procedure will be a full homography matrix, so four matches will be necessary at least to allow the computation, and $p = 4$ in the the LMedS implementation. To define the number of buckets, the following considerations have to be taken into account:

- Collinearity: If three or more of the four matches are collinear, then the system of equations is ill-posed and can't be solved.
- Matches very close together: This situation generates a correct homography computation, but only the local transformation is described by the resultant homography. The residue will be small for matches close to the ones used to

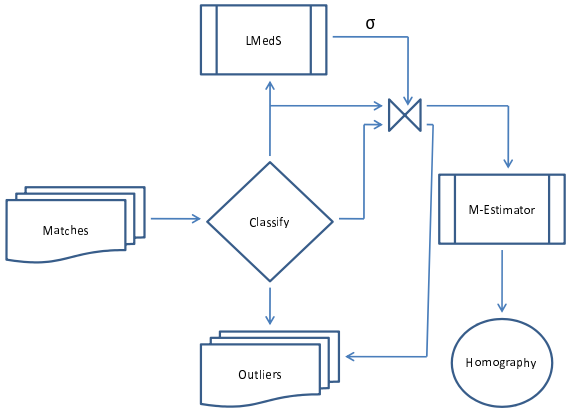


Figure 2.3: Homography computation diagram

Figure 2.4: Bucketing adopted for the LMedS implementation

compute the homography, but quite big for spatially far away matches. As a result, spatially far matches can be erroneously considered as outliers.

The solution adopted for the image bucketing is illustrated in Fig. 2.4. The image has been divided into eight areas of the same size, because matches must be as sparse as possible for a good homography estimation. This configuration allows to select eight consecutive matches from highly different image areas and, in general, avoids the selection of p matches close to each other. The criteria for bucket selection will be a random uniform variable from one to eight, and for match selection inside each bucket a random uniform variable from one to the number of matches in the bucket. This procedure guarantees random match selection.

2.3.2 Robust estimation based on M-Estimator

The M-Estimator ("M" for"maximum likelihood-type") method (Huber, 1981) is nowadays widely used for robust model fitting. The standard least-squares method tries to minimize $\sum_i r_i^2$ (where r_i is the residual of the i^{th} datum), which is unstable if there are outliers present in the data. The effect of outlaying data in the minimization process is so strong that the parameters thus estimated are distorted. The M-Estimator tries to reduce this effect by replacing the squared residual r_i^2 by another function of the residuals:

$$\min \sum_i \rho(r_i) \tag{2.17}$$

Where $\rho(x)$ is a symmetric, positive-definite function with a unique minimum at zero, and is chosen to be less increasing than square. Instead of solving directly this problem, an iterated reweighted least-squares is implemented.

Thus, let $\mathbf{p} = [p_1, ..., p_n]^T$ be the parameter vector to be estimated. The M-Estimator of $\mathbf{p}$ based on the function $\rho(r_i)$ is the vector $\mathbf{p}$ which is the solution of the following set of m equations:

$$\sum_i \psi(r_i) \frac{\partial r_i}{\partial p_j} = 0, \quad \text{for } j = 1, ..., n \tag{2.18}$$

Where the derivative $\psi(x) = \mathrm{d}\rho(x)/\mathrm{d}x$ is called the influence function. If the weight function is defined as:

$$w(x) = \frac{\psi(x)}{x} \tag{2.19}$$

Then (2.18) becomes:

$$\sum_i w(r_i) r_i \frac{\partial r_i}{\partial p_j} = 0, \quad \text{for } j = 1, ..., n \tag{2.20}$$

This is exactly the system of equations that is obtained if the following iterated reweighted least-squares problem is solved:

$$\min \sum_i w(r_i^{k-1}) r_i^2 \tag{2.21}$$

type	$\rho(x)$	$\psi(x)$	$w(x)$
L_1	$\lvert x\rvert$	$sgn(x)$	$\frac{1}{\lvert x\rvert}$
L_2	$\frac{x^2}{2}$	x	1
$L_1 - L_2$	$2(\sqrt{1+x^2/2}-1)$	$\frac{x}{\sqrt{1+x^2/2}}$	$\frac{1}{\sqrt{1+x^2/2}}$
L_p	$\frac{\lvert x\rvert^v}{v}$	$sgn(x)\lvert x\rvert^{v-1}$	$\lvert x\rvert^{v-2}$
$Fair$	$c^2\left(\frac{\lvert x\rvert}{c} - log\left(1+\frac{\lvert x\rvert}{c}\right)\right)$	$\frac{x}{1+\lvert x\rvert/c}$	$\frac{1}{1+\lvert x\rvert/c}$
$Cauchy$	$\frac{c^2}{2}log\left(1+(x/c)^2\right)$	$\frac{x}{1+(x/c)^2}$	$\frac{1}{1+(x/c)^2}$

Table 2.1: Example of weight functions for the M-Estimator

Where the superscript k indicates the iteration number. The weight $w(r_i^{k-1})$ should be recomputed after each iteration in order to be used in the next iteration. Several weight functions could be used depending on the data set and the model to be fitted, Table 2.1 summarizes some usual functions.

It is important to remark that the M-Estimator is not specially suitable for solving systems of equations with local minimums. This problem is easily solved by starting the iterations with a solution sufficiently close the global minima. The actual implementation of the homography computation pre-filters the data set and provides an initial solution by means of the LMedS algorithm, so it is expected a good behavior in the M-Estimator.

Implementation

The following equation defines the selected weighting function for the M-Estimator:

$$\omega(x) = \frac{1}{1+|x|/c} \tag{2.22}$$

The growth of the error x for the i^{th} equation implies a reduction in the weight $w(x)$. This weight multiplies the i^{th} equation of the system, the greater the error is, the smaller its influence on the complete system of equations.

Given the iterative nature of the algorithm, it is necessary to define an end-condition. The condition selected for this implementation is based not only on the

evolution of the estimated parameters but also on the evolution of the weights. Thus, the two following conditions must be met:

- The sum of the absolute variation of the estimated homography parameters with respect to the previous computation is lower than 10^{-3}.

  Given the homography computed in the current iteration $\mathbf{H}_j$ and the homography computed in the previous iteration $\mathbf{H}_{j-1}$ whose elements will be denoted as h_j^i and h_{j-1}^i from $i = 1$ to $i = 8$ (remember $h_j^9 = h_{j-1}^9 = 1$), the following condition holds:

$$k_h = \sum_{i=1}^{8} abs(h_j^i - h_{j+1}^i) < 10^{-3} \tag{2.23}$$

- The sum of the absolute variation of the computed weights with respect to the computed in the previous iteration is lower than 10^{-2}.

  Given a system of n equation, w_j^i is defined as the weight associated to equation i at iteration j. The following condition holds:

$$k_w = \sum_{i=1}^{n} abs(w_j^i - w_{j+1}^i) < 10^{-2} \tag{2.24}$$

In addition, it could be possible that the data set generates an inconsistent system of equations. In such case the above end-conditions couldn't be satisfied and the algorithm would never stop iterating. This problem is addressed by including a maximum number of iterations that, for this implementation, has been fixed to fifteen.

2.4 Homographies and uncertainties

Once the homography is computed, it is necessary to obtain a measure of the estimation accuracy. In this approach, the covariance matrix is used for this purpose. By rearranging the homography matrix into the following vector:

$$\mathbf{h} = \begin{bmatrix} h_{11} & h_{11} & \dots & h_{33} \end{bmatrix}^T \tag{2.25}$$

The covariance is represented by a 9×9 matrix with the variance of these variables in the diagonal and the cross-variances in the upper and lower triangles.

In (Hartley and Zisserman, 2004) it is proposed a method to compute the covariance matrix of this estimation. Thus, given a set of n matches:

$$\begin{gathered} S_m = \{\{\mathbf{m}_1, \mathbf{m}'_1\}, \{\mathbf{m}_2, \mathbf{m}'_2\}, ...\{\mathbf{m}_n, \mathbf{m}'_n\}\} \\ \mathbf{m}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \mathbf{m}'_i = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} \end{gathered} \tag{2.26}$$

the method can be summarized in the following 3 steps:

1. Obtain the Jacobian $\mathbf{J}$ of the transformation from $\mathbf{m}$ to $\mathbf{m}'$ with respect to the 9 parameters of the homography $\mathbf{h}$, evaluated at the estimated homography by the previous procedure. This Jacobian is a $2n \times 9$ matrix derived from the homographic relation between the n matches:

$$\begin{aligned} x'_i &= (h_{00}x_i + h_{01}y_i + h_{02})/(h_{20}x_i + h_{21}y_i + h_{22}) \\ y'_i &= (h_{10}x_i + h_{11}y_i + h_{12})/(h_{20}x_i + h_{21}y_i + h_{22}) \end{aligned} \tag{2.27}$$

2. Compute the covariance of the error for each match used to compute the homography (this is a $2x2$ matrix denoted by $\mathbf{C}_{\mathbf{m}_i}$). Assuming that the errors in the matches are uncorrelated, the diagonal matrix is created with the contribution of each match:

$$\mathbf{C}_{\mathbf{m}} = diag(\mathbf{C}_{\mathbf{m}_1}, \mathbf{C}_{\mathbf{m}_2}, \ldots, \mathbf{C}_{\mathbf{m}_n}) \tag{2.28}$$

3. Once $\mathbf{J}$ and $\mathbf{C}_{\mathbf{m}}$ are known, compute the homography covariance as:

$$\mathbf{C}_{\mathbf{h}} = (\mathbf{J}^T {\mathbf{C}_{\mathbf{m}}}^{-1} \mathbf{J})^{-1} \tag{2.29}$$

The unknown parameters in this algorithm are the error covariance for each match ($\mathbf{C}_{\mathbf{m}_i}$).

A rather straightforward approach could be to calculate the variance of the residue between $\mathbf{m}'_i$ and $\hat{\mathbf{m}}'_i = \mathbf{H}\mathbf{m}_i$ (the estimation of the match by means of the computed

Figure 2.5: Image bucketing carried out to improve the covariance estimation.

homography) and use it as error covariance for all the matches, so it is assumed that the error is approximately the same in the whole image. This method works well when the estimated homography is accurate, but usually fails when the estimation is poor, because the error may not be uniform.

To improve the calculation of the covariances, a bucketing technique is proposed. The key idea is to divide the image in several sections (see figure 2.5) and compute the variance of the residue separately in each area. The bucketing pattern takes into account the properties of the homographic transformations. In this way, each $\mathbf{C}_{\mathbf{m}_i}$ is the variance of the residue computed in the area to which the match i belongs.

2.5 Conclusions

The Chapter summarizes the Projective Geometry theory and describes a method to compute the homography matrix from a set of matches in a robust manner. Several practical issues are considered in this algorithm. The homography will be used in next Chapters as basic tool for real camera motion computation.

Although no experimental results are explicitly shown in this Chapter, this method has been used in all experiments carried out in the next four chapters and it has been successfully tested during the COMETS project (Ollero et al., 2005) for on-line compensation of the induced motion in the images by UAVs vibrations (Merino et al., 2006a).

Chapter 3

Homography-based odometry for UAVs

This chapter proposes a monocular visual odometer to act as a backup system when the accuracy of GPS is reduced to critical levels. The technique makes use of the homography computed between two consecutive images to estimate the real motion that the camera undergoes and, hence, the UAV. The technique avoids explicit 3D reconstruction, which allows to reduce the computational load and UAV payload given the baseline limitations.

The chapter is structured as follows. Firstly, the Chapter is introduced. Then, the geometry associated to multiple views of the same plane is detailed. Later, the fundamentals of the relative motion computation based on homographies and the algorithm is described. Finally some experimental results and discussions are shown.

3.1 Introduction

Related work on visual odometry for UAVs has been previously described in Chapter 1. Summarizing, monocular vision has been barely used for odometry by the research community. Most approaches rely on stereo vision for relative motion computation such as (Amidi et al., 1998), (Byrne et al., 2006) or (Lacroix et al., 2002),

while monocular vision is actively used for landing or take-off in planar targets as in (Shakernia et al., 2002), (Saripalli et al., 2003) or (Saripalli and Sukhatme, 2003).

The technique that will be presented in this chapter is based in monocular vision to avoid the restrictions that the base-line of a stereo vision system imposes when the UAV flies at medium/hight altitude.

No beacons or known landmarks will be used to localize the robot, only the image to image homography and the distance to the ground will be used to extract the real motion that the camera undergoes.

The homography will be computed from a set of matches between both images. The general purpose feature tracking algorithm described in (Ollero et al., 2004) provides these matches. Then, it is assumed that the scene has sufficient texture to allow tracking.

3.2 Geometry of two views of the same plane

Consider the position and orientation of two cameras in the world coordinate frame as shown in Fig. 3.1. Given the two projections $\mathbf{m_1} \in \mathbb{R}^2$ and $\mathbf{m_2} \in \mathbb{R}^2$ of a fixed point $\mathbf{P} \in \mathbb{R}^3$ belonging to a plane $\mathbf{\Pi}$, this section will detail the geometric relations induced in the two views of the same plane $\mathbf{\Pi}$.

Without loss of generality, it will be considered that the world coordinate frame is fixed to the camera one; that is, the translation with respect the origin is $\mathbf{t}_1 = [0, 0, 0]^T$ and the orientation is defined by the unit rotation matrix $\mathbf{R}_1 = \mathbf{I}$. Thus, if the Euclidean distance from the camera one to the plane $\mathbf{\Pi}$ is d_1 and the normal plane is defined by $\mathbf{n}_1$:

$$\mathbf{\Pi} : d_1 - \mathbf{n}_1^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.1}$$

If the calibration matrix $\mathbf{A}_1$ of the camera one is known, $\mathbf{m}_1$ can be back-projected into a straight line that crosses the origin with the direction $\mathbf{v}_1$, given by:

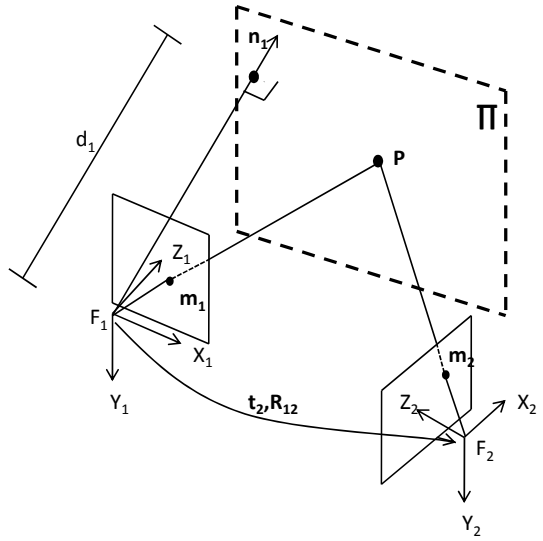


Figure 3.1: Geometry of two views of the same plane

$$\mathbf{v}_1 = \mathbf{A}_1^{-1}\tilde{\mathbf{m}}_1 \tag{3.2}$$

Where $\tilde{\mathbf{m}}_1 = [\lambda \mathbf{v}_1^T, \lambda]^T$ denotes the homogeneous representation of $\mathbf{v}_1$ when $\lambda \neq 0$. Then, knowing that $\mathbf{P}$ is the intersection of the back-projection of $\mathbf{m}_1$ and $\mathbf{\Pi}$:

$$d_1 - \mathbf{n}_1^T \mathbf{A}_1^{-1}\tilde{\mathbf{m}}_1 = 0 \tag{3.3}$$

Rearranging the equation:

$$\mathbf{A}_1^{-1}\tilde{\mathbf{m}}_1 = d_1 \mathbf{n}_1 \tag{3.4}$$

Thus, if the relative position and orientation of the two cameras, and both camera calibrations $\mathbf{A}_1$ and $\mathbf{A}_2$ are known, it is possible to relate the back-projection of $\mathbf{m}_1$ and $\mathbf{m}_2$ by:

$$\mathbf{v}_2 = \mathbf{R}_{12}(\mathbf{v}_1 - \mathbf{t}_2) \tag{3.5}$$

Where $\mathbf{R}_{12}$ is the rotation matrix that transforms the orientation of camera one into the orientation of camera two and $\mathbf{t}_2$ is the translation of camera two with respect to camera one expressed in the coordinate frame of camera one (see Fig. 3.1). By rearranging terms we get:

$$\mathbf{v}_2 = \mathbf{R}_{12}\left(\mathbf{I} - \frac{\mathbf{t}_2\mathbf{v}_1^T}{|\mathbf{v}_1|^2}\right)\mathbf{v}_1 \tag{3.6}$$

Taking into account equation (3.4) and replacing $\mathbf{v}_1$ and $\mathbf{v}_2$ by their value:

$$\tilde{\mathbf{m}}_2 = \mathbf{A}_2\mathbf{R}_{12}\left(\mathbf{I} - \frac{\mathbf{t}_2\mathbf{n}_1^T}{d_1}\right)\mathbf{A}_1^{-1}\tilde{\mathbf{m}}_1 \tag{3.7}$$

Therefore, it is possible to link the projection of set of points $\{\mathbf{P}_0, \mathbf{P}_1, ...\mathbf{P}_n\}$ in two different cameras if the set lays in a plane. In fact, for this particular case, the transformation between the features $\mathbf{m}_1$ and $\mathbf{m}_2$ is a plane-to-plane homography, so:

$$\tilde{\mathbf{m}}_2 = \mathbf{H}_{12}\tilde{\mathbf{m}}_1 \tag{3.8}$$

Thus, if the cameras are viewing the same planar scene and we are able to compute the homography that relates the two views by using the techniques described in Chapter 2, it could be possible to extract the rotation, normal and translation between the cameras:

$$\mathbf{H}_{12} = \mathbf{A}_2\mathbf{R}_{12}\left(\mathbf{I} - \frac{\mathbf{t}_2\mathbf{n}_1^T}{d_1}\right)\mathbf{A}_1^{-1} \tag{3.9}$$

However, given the bearing only nature of the camera, the translation will be defined up to the scale factor $1/d_1$. To completely recover the motion it will be necessary to measure the distance from camera one to plane $\mathbf{\Pi}$. This information can be easily recovered by means of different methods such as range sensor, barometric sensor or even the GPS last reliable altitude measure.

Finally, notice that the problem can be reformulated as a single camera whose position and orientation change throughout time. In this case the calibration matrix is the same for both views, so $\mathbf{A}_1 = \mathbf{A}_2$. Thus, given a set of matches between two

sequenced images and a planar scene, the induced homography combined with a range sensor can be used to compute the real motion that the camera undergoes.

3.3 Rotation, translation and plane normal computation

This section will deal with the computation of the rotation, translation and plane normal included in the homography that relates two views of the same planar scene. A solution based on the singular value decomposition (SVN) of the homography will be used.

3.3.1 Homography decomposition

Consider a single camera that moves throughout time, the homography $\mathbf{H}_{12}$ that relates the first and the second view of the same planar scene and the camera calibration matrix $\mathbf{A}$. According to equation (3.9), the uncalibrated homography is defined as:

$$\mathbf{H}_{12}^{u} = \mathbf{A}^{-1}\mathbf{H}_{12}\mathbf{A} = \mathbf{R}_{12}\left(\mathbf{I} - \frac{\mathbf{t}_2\mathbf{n}_1^T}{d_1}\right) \tag{3.10}$$

$\mathbf{H}_{12}^{u}$ can be decomposed into singular values and then be rewritten as $\mathbf{H}_{12}^{u} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are 3×3 rotation matrices and $\mathbf{D} = diag(\lambda_1, \lambda_2, \lambda_3)$ a diagonal matrix whose elements satisfy that $\lambda_1^2, \lambda_2^2, \lambda_3^2$ are the eigenvalues of $\mathbf{H}_{12}^{u}(\mathbf{H}_{12}^{u})^T$. The columns of $\mathbf{U}$ and $\mathbf{V}$ will be denoted as $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ and $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ respectively.

Once $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{D}$ have been conveniently ordered such us $\lambda_1 > \lambda_2 > \lambda_3$, the values of the diagonal matrix $\mathbf{D}$ can be used to distinguish three types of movements carried out by the camera (Tsai et al., 1982):

- The three singular values of $\mathbf{H}_{12}^{u}$ are equal, so $\lambda_1 = \lambda_2 = \lambda_3$. It occurs when the motion consist of rotation around an axis through the origin only, i.e., $\mathbf{t}_2 = \mathbf{0}$. The rotation matrix is unique and can be computed as:

$$\mathbf{R}_2 = \frac{1}{\lambda_1}\mathbf{H}_{12}^{u} \tag{3.11}$$

In this case there is not sufficient information to estimate the plane normal $\mathbf{n}_1$.

- The multiplicity of the singular values of $\mathbf{H}_{12}^u$ is two, for example $\lambda_1 = \lambda_2 \neq \lambda_3$. Then, the solution for motion and geometrical parameters is unique up to a common scale factor for the translation parameters:

$$\begin{aligned} \mathbf{R}_2 &= \tfrac{1}{\lambda_1}\mathbf{H} - \left(\tfrac{\lambda_3}{\lambda_1} - s\right)\mathbf{u}_3\mathbf{v}_3^T \\ \mathbf{t}_2 &= \tfrac{1}{w}\left(\tfrac{\lambda_3}{\lambda_1} - s\right)\mathbf{u}_3 \\ \mathbf{n}_1 &= w\mathbf{v}_3 \end{aligned} \tag{3.12}$$

Where $s = det(\mathbf{U})det(\mathbf{V})$ and w is a scale factor.

- The three singular values of $\mathbf{H}_{12}^u$ are different, i.e., $\lambda_1 \neq \lambda_2 \neq \lambda_3$. In this case two possible solutions for rotation, translation and plane normal exist and can be computed as:

$$\begin{aligned} \mathbf{R}_2 &= \mathbf{U}\begin{bmatrix} \alpha & 0 & \beta \\ 0 & 1 & 0 \\ -s\beta & 0 & s\alpha \end{bmatrix}\mathbf{V}^T \\ \mathbf{t}_2 &= \tfrac{1}{w}\left(-\beta\mathbf{u}_1 + \left(\tfrac{\lambda_3}{\lambda_2} - s\alpha\right)\mathbf{u}_3\right) \\ \mathbf{n}_1 &= w(\delta\mathbf{v}_1 + \mathbf{v}_3) \end{aligned} \tag{3.13}$$

Where:

$$\begin{aligned} \delta &= \pm\sqrt{\frac{\lambda_1^2 - \lambda_2^2}{\lambda_2^2 - \lambda_3^2}} \\ \alpha &= \frac{\lambda_1 + s\lambda_3\delta^2}{\lambda_2(1+\delta^2)} \\ \beta &= \pm\sqrt{1-\alpha^2} \\ s &= det(\mathbf{U})det(\mathbf{V}) \end{aligned}$$

Each solution must accomplish that $sgn(\beta) = -sgn(\delta)$.

The presence on noise in both feature tracking and homography estimation always leads to different singular values for $\mathbf{H}_{12}^{u}$ and the third of the previous cases becomes the dominant in real conditions. For the third case, a robust algorithm is presented in (Triggs, 1998). This method has been implemented and tested, and the results will be shown in Section 3.4.

However, two possible solutions $\{\mathbf{R}_2^1, \mathbf{t}_2^1, \mathbf{n}_1^1\}$ and $\{\mathbf{R}_2^2, \mathbf{t}_2^2, \mathbf{n}_1^2\}$ will be obtained, so an additional step will be needed in order to identify the correct solution. For this purpose the next section proposes a method based on the uniqueness of the plane normal.

3.3.2 Correct solution disambiguation

In (Shakernia et al., 2002) a method to compute the correct solution based on the uniqueness of the normal $\mathbf{n}$ is proposed; that is, the variation of the normal plane from a correct solution to the next one should be smooth given that the plane will be approximately the same when images are captured at a sufficient high rate. Thus, given the set of possible normals:

$$S_n = \{\mathbf{n}_1^1, \mathbf{n}_1^2, \mathbf{n}_2^1, \mathbf{n}_2^2, \mathbf{n}_3^1, \mathbf{n}_3^2, ...\} \tag{3.14}$$

where the superindex denotes the two possible normal solutions and the subindex is the image number sequence. This method defines an empirical threshold ϵ, and considers as the correct solution the normal $\mathbf{n}_1^1$ or $\mathbf{n}_1^2$ that allows to find a sequence of normals that meets:

$$\|\mathbf{n}_1^{1,2} - \mathbf{n}_i^{1,2}\| \leq \epsilon \quad \forall i > 1 \tag{3.15}$$

where ϵ must be carefully tuned to guarantee a proper algorithm operation. If the value of ϵ is too high then both solutions can meet the constraint and it is not possible to detect the correct one. On the other hand, if ϵ is too low then no solution would be valid.

Algorithm 1 Unique normal computation

for $i = 1$ to 2 **do**
  Compute the distances among $\mathbf{n}_1^i$ and the rest of normals of S_n.
  $\epsilon_i = 0.5$
  solution = check_uniqueness(ϵ_i,S_n)
  while solution $\neq$ UNIQUE **do**
    if solution = MULTIPLE **then**
      $\epsilon_i = \epsilon_i - 0.1$
    else
      $\epsilon_i = \epsilon_i + 0.1$
    end if
    solution = check_uniqueness(ϵ_i,S_n)
  end while
end for

In the following, a more reliable method to detect the correct solution is proposed. If $\mathbf{n}_1^1$ and $\mathbf{n}_1^2$ were correct, there would be two set of solutions, S_{n^1} and S_{n^2}. The uniqueness of the normal leads to the following constraints:

$$\|\mathbf{n}_1^1 - \mathbf{n}_j^i\| \leq \epsilon_1 \quad \forall \mathbf{n}_j^i \in S_{n^1} \tag{3.16}$$

$$\|\mathbf{n}_1^2 - \mathbf{n}_j^i\| \leq \epsilon_2 \quad \forall \mathbf{n}_j^i \in S_{n^2} \tag{3.17}$$

where ϵ_1 and ϵ_2 are the minimal values that guarantee an unique solution for equations (3.16) and (3.17) respectively. The pairs $\{S_{n^1}, \epsilon_1\}$ and $\{S_{n^2}, \epsilon_2\}$ are computed separately by means of the Algorithm 1. Then, the correct solution is chosen between both options as the one that achieves the minimum ϵ.

3.3.3 An estimation of the uncertainties

This section deals with the computation of the uncertainties committed to the estimated rotation, translation and plane normal given the covariance matrix associated to the homography (computed using the method detailed in section 2.4).

The proposed method will compute the Jacobian of the complete process to obtain a first order approximation of the rotation, translation and plane normal covariance

matrices. Particularly, once the uncalibrated homography has been decomposed into its singular values, the previous section showed how the computation of the camera motion is simple, so this section will focus in the computation of the Jacobian associated to the singular value decomposition process.

Thus, given the uncalibrated homography $\mathbf{H}_{12}^{u}$, it can be decomposed as follows:

$$\mathbf{H}_{12}^{u} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} = \mathbf{U}\mathbf{D}\mathbf{V}^{T} = \sum_{i=0}^{2} \left(\lambda_i \mathbf{u}_i \mathbf{v}_i^{T}\right) \tag{3.18}$$

The goal is to compute $\frac{\partial \mathbf{U}}{\partial h_{ij}}$, $\frac{\partial \mathbf{V}}{\partial h_{ij}}$ and $\frac{\partial \mathbf{D}}{\partial h_{ij}}$ for all the h_{ij} of the matrix $\mathbf{H}_{12}^{u}$.

Taking the derivative of equation (3.18) with respect to h_{ij} yields the following expression:

$$\frac{\partial \mathbf{H}_{12}^{u}}{\partial h_{ij}} = \frac{\partial \mathbf{U}}{\partial h_{ij}}\mathbf{D}\mathbf{V}^{T} + \mathbf{U}\frac{\partial \mathbf{D}}{\partial h_{ij}}\mathbf{V}^{T} + \mathbf{U}\mathbf{D}\frac{\partial \mathbf{V}^{T}}{\partial h_{ij}} \tag{3.19}$$

Clearly, $\forall (k,l) \neq (i,j)$, $\frac{\partial h_{kl}}{\partial h_{ij}} = 0$ while $\frac{\partial h_{ij}}{\partial h_{ij}} = 1$. Since $\mathbf{U}$ is an orthogonal matrix:

$$\mathbf{U}^{T}\mathbf{U} = \mathbf{I} \Rightarrow \frac{\partial \mathbf{U}^{T}}{\partial h_{ij}}\mathbf{U} + \mathbf{U}^{T}\frac{\partial \mathbf{U}}{\partial h_{ij}} = \mathbf{\Omega}_{U}^{ij\,T} + \mathbf{\Omega}_{U}^{ij} = 0 \tag{3.20}$$

Where $\mathbf{\Omega}_{U}^{ij}$ is defined by

$$\mathbf{\Omega}_{U}^{ij} = \mathbf{U}^{T}\frac{\partial \mathbf{U}}{\partial h_{ij}} \tag{3.21}$$

It is clear that $\mathbf{\Omega}_{U}^{ij}$ is an antisymmetric matrix.

Similarly, an antisymmetric matrix $\mathbf{\Omega}_{V}^{ij}$ can be defined for $\mathbf{V}$ as:

$$\mathbf{\Omega}_{V}^{ij} = \frac{\partial \mathbf{V}^{T}}{\partial h_{ij}}\mathbf{V} \tag{3.22}$$

By multiplying (3.19) by $\mathbf{U}^{T}$ and $\mathbf{V}$ from left and right respectively, and using (3.21) and (3.22), the following relation is obtained:

$$\mathbf{U}^{T}\frac{\partial \mathbf{H}_{\mathbf{12}}^{\mathbf{u}}}{\partial h_{ij}}\mathbf{V} = \mathbf{\Omega}_{U}^{ij}\mathbf{D} + \frac{\partial \mathbf{D}}{\partial h_{ij}} + \mathbf{D}\mathbf{\Omega}_{V}^{ij} \tag{3.23}$$

Since $\mathbf{\Omega}_U^{ij}$ and $\mathbf{\Omega}_V^{ij}$ are antisymmetric matrices, all their diagonal elements are equal to zero. Recalling that $\mathbf{D}$ is a diagonal matrix, it is easy to see that the diagonal elements of $\mathbf{\Omega}_U^{ij}\mathbf{D}$ and $\mathbf{D}\mathbf{\Omega}_V^{ij}$ are also zero. Thus:

$$\frac{\partial \lambda_k}{\partial h_{ij}} = u_{ik} v_{jk} \tag{3.24}$$

Taking into account the antisymmetric property, the elements of the matrices $\mathbf{\Omega}_U^{ij}$ and $\mathbf{\Omega}_V^{ij}$ can be computed by solving a set of $2x2$ linear systems, which are derived from the off-diagonal elements of the matrices in (3.23):

$$\left.\begin{array}{rcl} d_l \mathbf{\Omega}_{U\,kl}^{ij} + d_k \mathbf{\Omega}_{V\,kl}^{ij} & = & u_{ik} v_{jl} \\ d_k \mathbf{\Omega}_{U\,kl}^{ij} + d_l \mathbf{\Omega}_{V\,kl}^{ij} & = & -u_{il} v_{jk} \end{array}\right\} \tag{3.25}$$

Where the index ranges are $k = 0...2$ and $l = i + 1...2$. Note that, since the d_k are positive numbers, this system has a unique solution provided that $d_k \neq d_l$. Assuming for the moment that $\forall (k,l), d_k \neq d_l$, the 3 parameters defining the non-zero elements of $\mathbf{\Omega}_U^{ij}$ and $\mathbf{\Omega}_V^{ij}$ can be easily recovered by solving the 3 corresponding $2x2$ linear systems.

Once $\mathbf{\Omega}_U^{ij}$ and $\mathbf{\Omega}_V^{ij}$ have been computed, the partial derivatives are obtained as follows:

$$\frac{\partial \mathbf{U}}{\partial h_{ij}} = \mathbf{U}\mathbf{\Omega}_U^{ij} \tag{3.26}$$

$$\frac{\partial \mathbf{V}}{\partial h_{ij}} = -\mathbf{V}\mathbf{\Omega}_V^{ij} \tag{3.27}$$

Taking into account the (3.24), (3.26) and (3.27) and the covariance matrix corresponding to the homography it is possible to compute the covariance matrix associated to $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{D}$. Further details and demonstrations can be found in (Papadopoulo and Lourakis, 2000).

Finally, the Jacobians of the linear algebra used to extract the rotation, translation and normal are easily computed and combined with these covariances to compute the final covariances matrix.

(a)

(b)

Figure 3.2: HERO helicopter.

3.4 Experimental results

This section will show some experimental results in which the homography-based visual odometer is applied to monocular image sequences gathered by a real UAV, the HERO helicopter. The results presented here have been validated using the navigation sensors on board the UAV.

The visual odometer algorithm (feature tracking, robust homography computation and homography decomposition) has been programmed in C++ code and run in a Intel Centrino Duo Laptop at $10Hz$ with $320x240$ images.

Next paragraphs will introduce the HERO helicopter and the experiments carried out. Then, two experiments will be detailed, a short image sequence and a longer one with a sharp movement.

3.4.1 The HERO helicopter

HERO is an aerial robotic platform designed for research on UAV control, navigation and perception, see Fig. 3.2. It has been developed by the "Robotics, Vision and Control Research Group" at the University of Seville during the CROMAT project, funded by the Spanish Government.

HERO is equipped with accurate sensors to measure position and orientation, cameras and a PC-104 to allow processing on board. The sensors and processing is

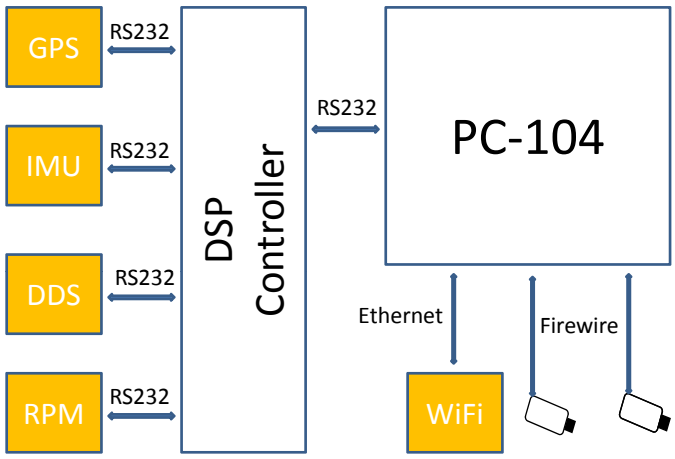


Figure 3.3: Hardware and sensor architecture in HERO helicopter

distributed as shown in Fig. 3.3. Basically, a DSP is used as data acquisition system and low level controller (position and orientation), the PC-104 runs the rest of tasks such us perception, communications or navigation. All the data gathered by the DSP are exported to the PC-104 through a serial line and published for the rest of the processes.

All the sensor data have been logged together with the images in order to avoid inconsistency among different sensor data. The position is estimated with a Novatel DGPS with $2cm$ accuracy and updated at $5Hz$ while the IMU provide orientation at $50Hz$ with accuracy os 0.2 degrees. In the experiments show in next sections, the camera was oriented forty-five degrees with respect to the helicopter horizontal.

3.4.2 A short experiment

In this experiment the UAV softly moved from the starting point to the right and then went back approximately to the initial position. It is a sequence composed by 250 images, or approximately 25 seconds of flight. The estimated position by using the visual odometer is shown in Fig. 3.4. The figure presents the DGPS position estimation and the errors committed by the odometer. It can be seen that the errors in X, Y and Z axis are shorter than 2 meters. In addition, the estimated standard deviation is coherent in three axis.

It is important to notice that the odometry is computed taking into account the estimated translation and rotation so it accumulates both errors.

Figure 3.5 shows the evolution of the estimated orientation by using the odometer and the helicopter IMU. The orientation has been represented in quaternions such as $\mathbf{q} = [q_x, q_y, q_z, q_w]^T$ with norm one, where $\mathbf{v} = [q_x, q_y, q_z]^T$ is the rotation vector and q_w is the rotation angle (see Appendix A). It can be seen in Fig. 3.5 that the error in the orientation estimation are small and the standard deviation is overall consistent except for q_z where is slightly subestimated.

3.4.3 A longer experiment

The experiment were performed in the same conditions as the above one. The sequence is composed by 650 images, or approximately 65 seconds of flight. Here, a sharp movement is made around the image samples 400 as shown in Fig. 3.7(c) and Fig. 3.7(e) due to an error in the UAV control loop.

The estimated position by using the visual odometer is shown in Fig. 3.6. The figure presents the DGPS position estimation and the errors associated to the odometer. It can be seen how the errors grow with the image samples. The errors corresponding to each estimation is added to the previous ones and make the position estimation diverge through time. It can be seen how the estimation of the standard deviation is coherent with the evolution of the error.

Figure 3.7 shows the evolution of the estimated orientation by using the odometer and the helicopter IMU expressed in quaternions. It can be seen in Fig. 3.7 how the errors in the estimated orientation are small but, again, the estimated standard deviation for q_z is subestimated. Possibly, the linear approximation in the Jacobian is inaccurate. To avoid this problem a second order approximation for the covariance matrix could be implemented using the iterative method proposed in (Julier and Uhlmann, 1997).

3.5 Conclusions

This chapter presents a method for UAV position estimation based on monocular image processing which can be considered as a visual odometer. The method is robust and could be easily implemented to provide real-time position information of the UAV. The experimental results show the feasibility of the approach and demonstrate that it can be used as an effective backup of GPS.

However, the accumulative error intrinsic to the odometry estimation make the position diverge through time. If a consistent view of the environment could be built, the odometry error could be compensated and thus the applicability of the technique would increase. Next Chapter will deal with this issue.

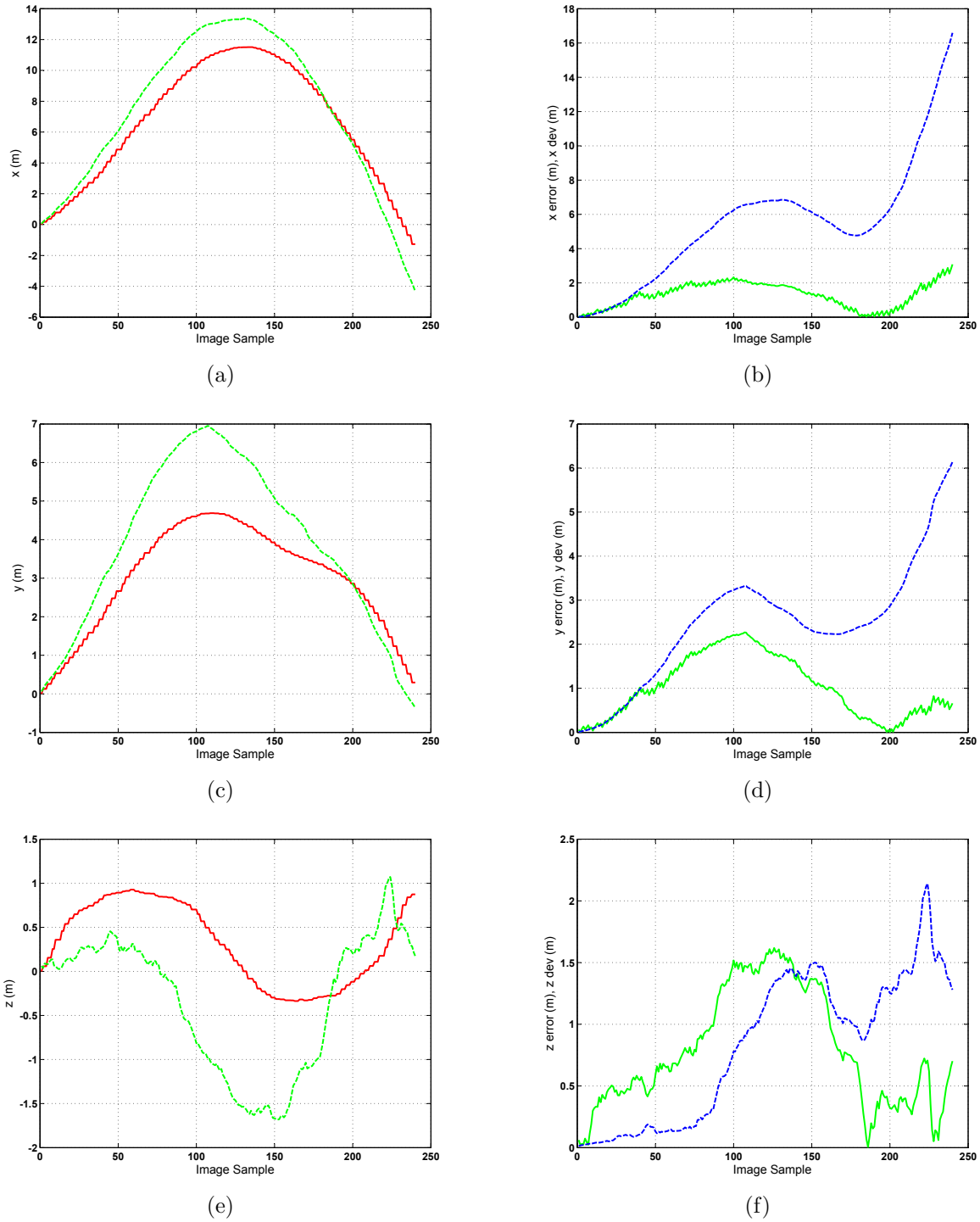


Figure 3.4: Left: Position estimation using vision based technique (green dashed line) and DGPS estimation (red solid line). Right: Error of the vision based odometry (green solid line) and estimated standard deviation (blue dashed line).

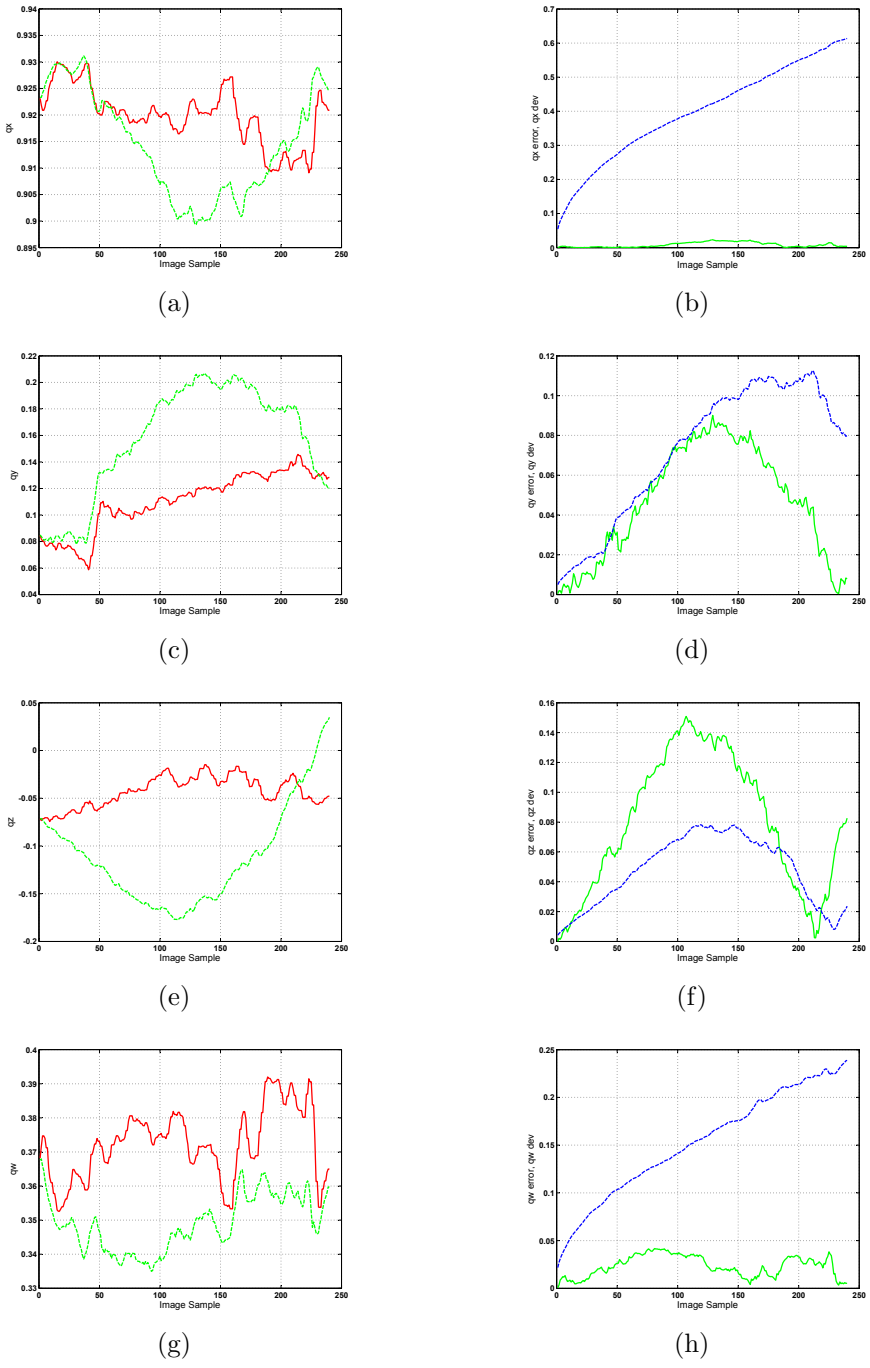


Figure 3.5: Left: Estimated orientation by using vision based technique (green dashed line) and IMU estimation (red solid line). The orientation is represented in quaternion. Right: Errors in the vision based estimation (green solid line) and estimated standard deviation (blue dashed line).

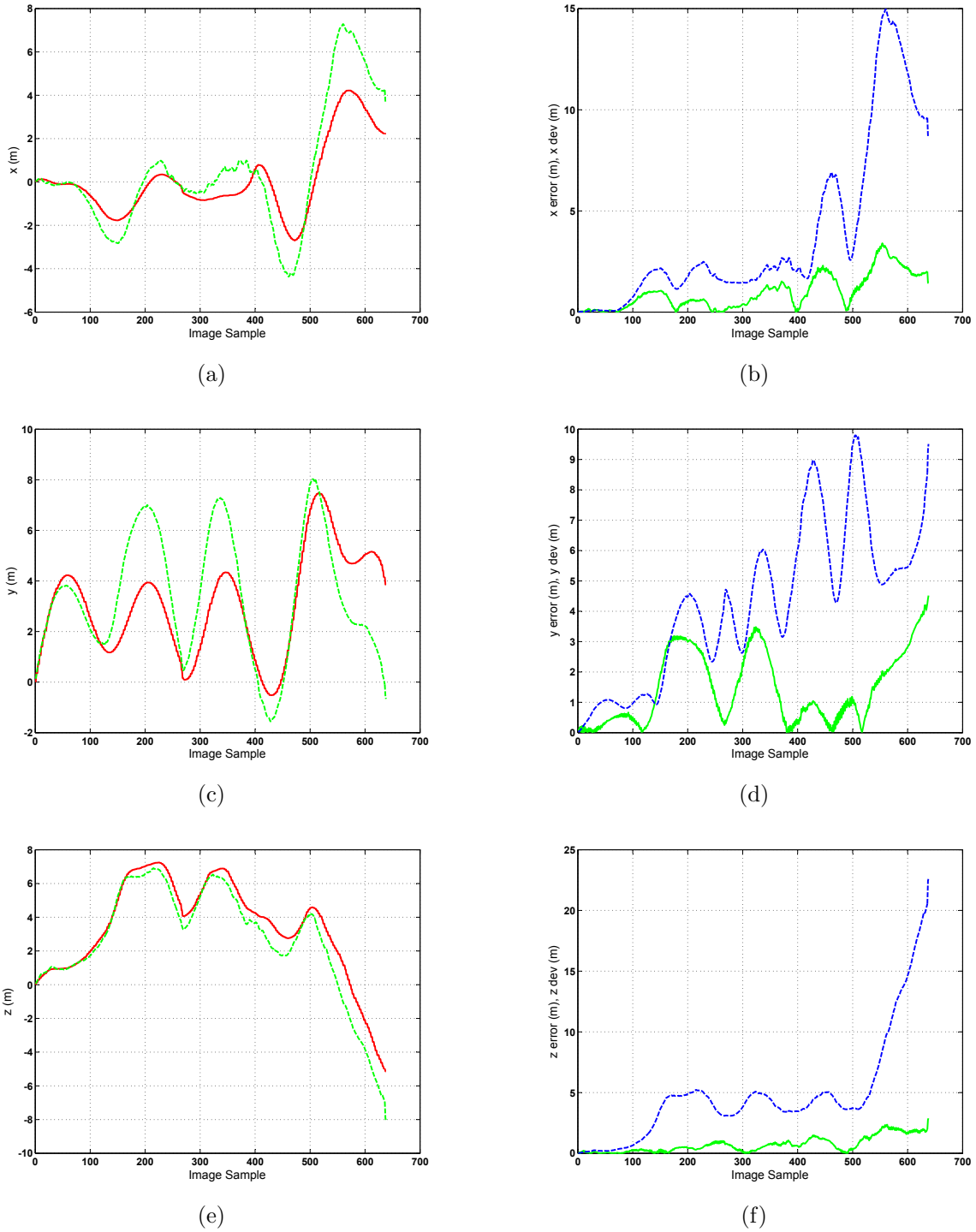


Figure 3.6: Left: Position estimation using vision based technique (green dashed line) and DGPS estimation (red solid line). Right: Error of the vision based odometry (green solid line) and estimated standard deviation (blue dashed line).

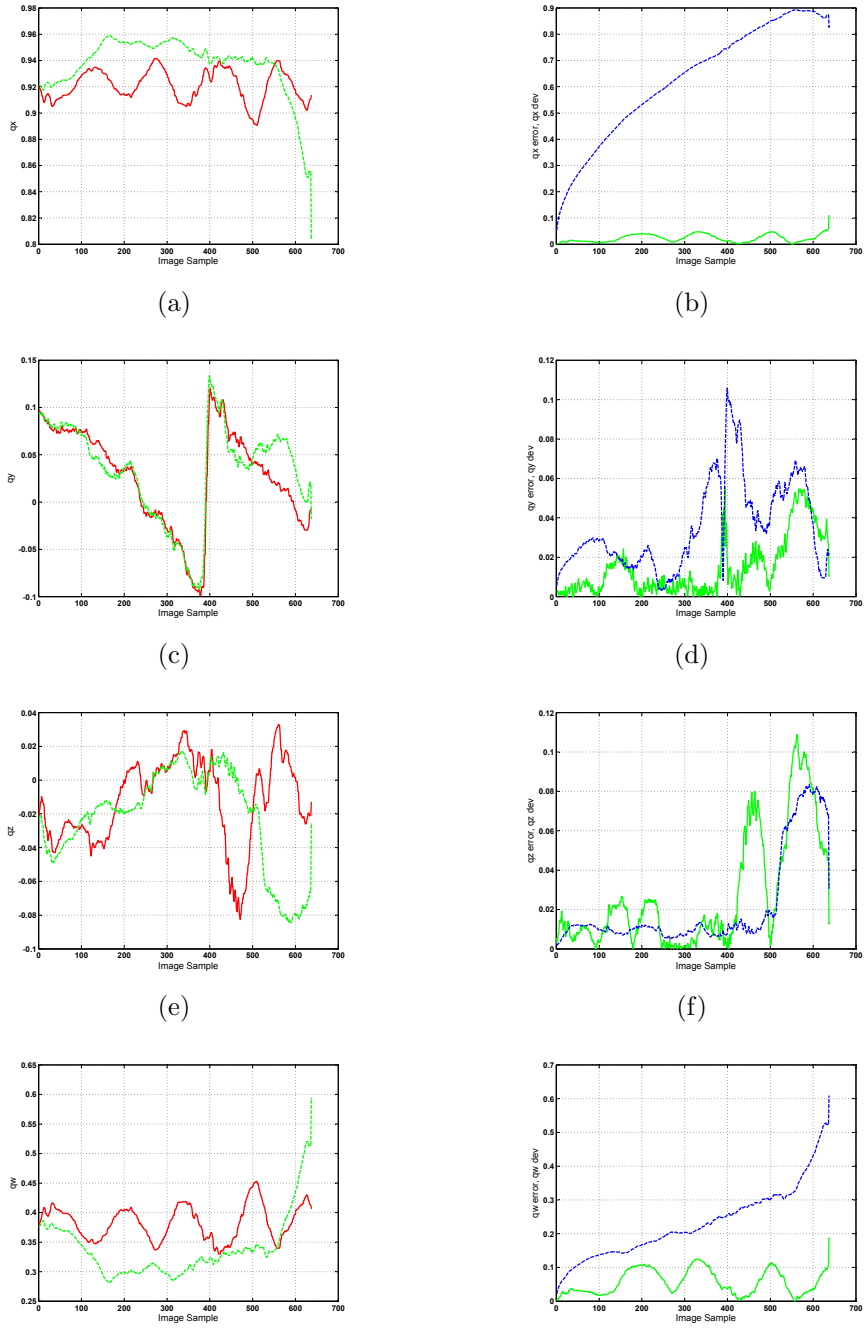


Figure 3.7: Left: Estimated orientation by using vision based technique (green dashed line) and IMU estimation (red solid line). The orientation is represented in quaternion. Right: Errors in the vision based estimation (green solid line) and estimated standard deviation (blue dashed line).

Chapter 4

UAV localization based on online mosaicking

It has been shown how the odometry based on homographies could be a good estimation of the motion that a UAV undergoes when it flies at medium/high altitude, but accumulative errors in the estimation can make the position diverge. This Chapter proposes a method based on online-built mosaics to correct the drift associated to the homography based motion estimation algorithm.

Mosaics allow using not only the current image but also previously recorded information for localization. A mosaic can be built from monocular sequences of images gathered by the UAV by using the homographic relation induced by the planar terrain. It will store all the information gathered by the camera and will define the mapping relationship between images and the planar ground. Thus, when a previously visited part of the scene falls into the field of view of the vehicle, the image could be matched with the mosaic to detect and decrease the accumulated drift.

The Chapter is organized as follows: Section 4.1 presents related works and an overview of the technique for online mosaicking. Sections 4.2, 4.3 and 4.4 describe the technique used for mosaic building. Finally, Section 4.5 applies the mosaic to refine the motion estimation algorithm described in Chapter 3 and shows some experimental results.

4.1 Introduction

A mosaic can be defined as an artificial picture of a scene. This picture is composed by an ordered set of images of smaller size than the mosaic. The images can be significantly rotated and translated with respect to the reference frame, in this case the first image.

Mosaics are built on top of tools that allow to compute the relative displacement among all the images and then properly locate the images to create the whole picture. In general, the process can be divided in three steps:

Local and global alignment This step computes the apparent motion between the current image and the previous one. This motion is modeled and then composed with the previous ones in order to align the current image to the reference frame.

Global alignment error reduction The second step is carried out in order to decrease the positioning error of the images inside the mosaic. While the first step computes the relative motion without any consideration about the positioning in the mosaic or previous error in the computation of the relative motion, this stage tries to detect inconsistency between the position estimated using the composition of the relative motions and the position of the image in the mosaic built until now. If an inconsistence is detected, it will be modeled and compensated.

Mosaic updating Finally, this step integrates the images into the mosaic.

Some methods have been proposed for computing the relative displacement among images by using inertial sensors like IMU and GPS such as the techniques presented in (Schultz et al., 2002; Rzhanov et al., 2001). These methods usually introduce small errors in the global alignment of the images but need image processing for a good local alignment. Although these techniques are robust, there exist a growing interest in computing these relations only from image processing. This way, the estimated relations among the images of the mosaic are independent of the UAV inertial sensors and could be later used for sensor reliability or to refine the localization.

The main drawback of the image processing-based mosaics is the accumulated error introduced in the global alignment of the images. Recently, new iterative methods have been used to minimize these errors via topology inference, like (Hsu et al., 2002; Shum and Szeliski, 1998). A topological network is built with the images and then used to reduce the errors in the alignment among all the images in a minimization process. The method offers good results for batch processing but their iterative nature leads to processing time problems in online implementations.

The proposed approach tries to detect global alignment errors each time a new image is introduced in the mosaic. The method is low accurate when comparing to topological-based approaches, but allows an online implementation.

A similar approach has been proposed for autonomous underwater vehicles (AUV) in (Garcia et al., 2001). The underlying concept is the same: to use a mosaic as environment representation of a vehicle when it moves at relative high altitude with respect to the ground because, in this case, the scene can be fairly approximated by a plane. However, the image apparent motion only considers affine homographies and there is no a global alignment error reduction.

Figure 4.1 depicts the proposed method for online mosaicking. By using the information given by an interest point matching algorithm (Ferruz and Ollero, 2000; Ollero et al., 2004), the local alignment between consecutive images can be obtained. The local transformation is modeled by a homography, computed from the matches by using a robust outlier rejection procedure. By composing consecutive homographies, the current image can be transformed into a global frame, in which the mosaic is built. However, if the transformed frame overlaps the current mosaic, the common information will be used to refine this global alignment. Finally, this global transformation is used to compute the motion between the current camera position and the initial one.

4.2 Local and global alignment

The goal of this stage is to compute the relative motion between two consecutive images, and then, the transformation that aligns the current image with the reference

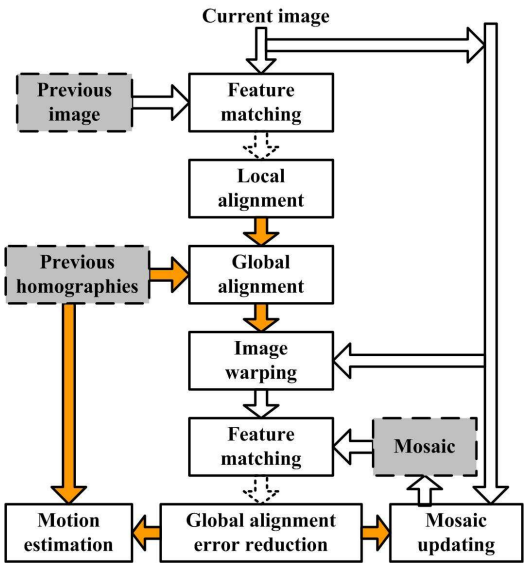


Figure 4.1: Overview of the method.

frame. First, the apparent image motion between consecutive images is extracted by means of a point-like feature matching function that provides a set of matches between the previous image and the current one. This image motion is modeled by a homography matrix using the techniques described in Chapter 2. The model holds when the camera undergoes a pure rotation or when the scene is planar, which is a good approximation when an UAV flies at high altitude.

Given the linear nature of the homographic transformations in homogeneous coordinates, it is possible to express the image motion from image n to the reference frame k, $\mathbf{H}_{kn}$, as a composition of the relative motion between the intermediate images. If T_I is a sequence of n images of a planar scene:

$$T_I = \{\mathbf{I}_0, \mathbf{I}_1, \mathbf{I}_2 ..., \mathbf{I}_n\} \tag{4.1}$$

and the $n-1$ homography matrixes that relate these images are known:

$$\mathbf{H}_{01}, \mathbf{H}_{12}, \mathbf{H}_{23} ..., \mathbf{H}_{(n-1)n} \tag{4.2}$$

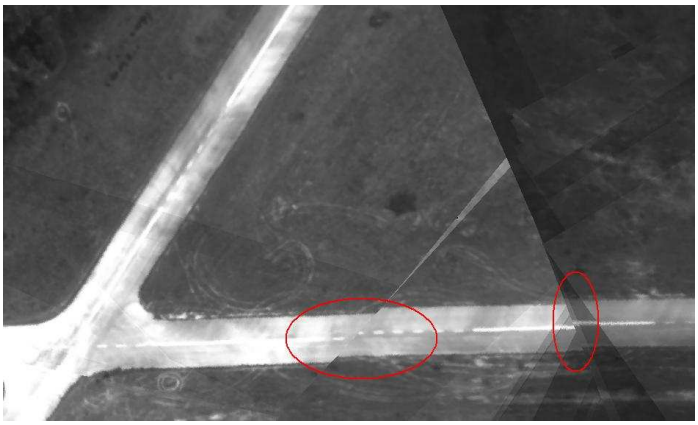


Figure 4.2: Piece of mosaic with global alignment errors (ellipses)

then, the transformation from image n to k will be defined by:

$$\mathbf{H}_{kn} = \prod_{i=k}^{n-1} \mathbf{H}_{i(i+1)} \tag{4.3}$$

Thus, it is possible to compose the complete motion between whatever image of the sequence and the reference frame by means of the relative motion computation. In the rest of the Chapter, the first image (image 0) of the sequence will be considered as reference frame.

The main drawback of this method is the accumulative positioning error of the images inside the mosaic. The homography estimation involves outlier rejection, least squares and robust model estimation over a set of matching candidates. Although by using the techniques presented in Chapter 2 the residual error of the computed homography is usually small, equation (4.3) leads to error accumulation. In addition, if the number of outliers is too high (due to matching errors, independent object motion or parallax effect) or the number of matches between images is small (blurred images or small image overlap) the residual error can be considerable.

Figure 4.2 shows a piece of mosaic which reflects the effects of the accumulative errors in the homography estimation. The errors are more evident when the mosaic is composed by images taken in large loops, and the accumulated registration errors generate visible inconsistences in the global image alignment.

Figure 4.3: Global alignment errors due to accumulative misregistration. The red square denotes the correct position inside the mosaic. The green square marks the position computed by using local alignment

4.3 Global alignment error reduction

The idea behind the proposed approach is to use the information stored in the mosaic to refine the global transformation from the current image to the common frame (transformation that will be used later to estimate the motion of the vehicle).

Thus, consider that at time i, the global alignment estimation procedure provides a homography that relates the current image $\mathbf{I}_i$ to the mosaic computed in the previous step $\mathbf{M}_{i-1}$. This homography is a noisy estimation that will be denoted as $\hat{\mathbf{H}}_{0i}$. The actual transformation will then be $\mathbf{H}_{0i}$ (assuming $\mathbf{I}_0$ is the reference frame).

Due to the accumulative drift at time i, the warped image $\mathbf{I}_i^w = \mathbf{H}_{0i}(\mathbf{I}_i)$ will be misaligned with respect to its correct position within the mosaic (see Fig. 4.3). If this alignment error is modeled by a full homography matrix called $\mathbf{H}_i^e$, then:

$$\mathbf{H}_{0i} = \mathbf{H}_i^e \hat{\mathbf{H}}_{0i} \tag{4.4}$$

The homography $\mathbf{H}_i^e$ can be estimated by matching the warped image $\mathbf{I}_i^w$ and the portion of the mosaic $\mathbf{M}_{i-1}$ where $\mathbf{I}_i^w$ should be placed. This matrix will encapsulate all alignment errors from frame 0 to i. Combining (4.3) and (4.4), the final alignment expression that aligns the image $\mathbf{I}_i$ to the mosaic $\mathbf{M}_{i-1}$ is given by:

$$\mathbf{H}_{0i} = \mathbf{H}_i^e \prod_{j=0}^{i} \mathbf{H}_{j(j+1)} \tag{4.5}$$

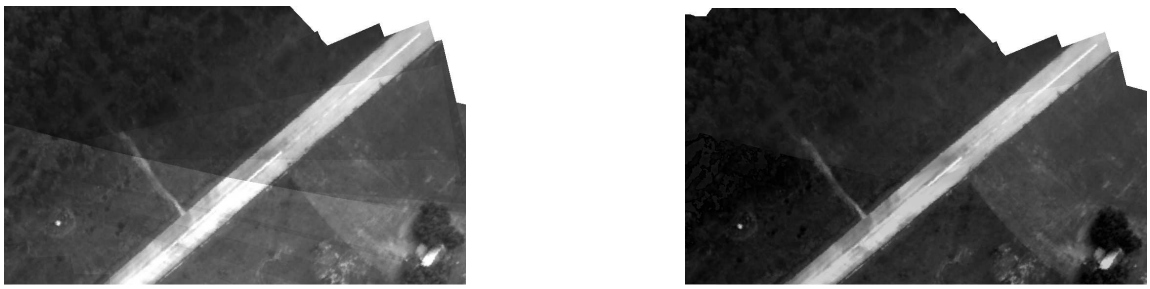
The alignment error reduction is carried out only when the UAV visits zones already stored in the mosaic and $\mathbf{H}_i^e$ will be computed if the estimated overlap between the image and the mosaic is greater than 50% of the image total size. Also, the number of correct matches between the warped image and the mosaic are considered; if this number is below a threshold, the correction will not be used because in such a case the estimation is too poor to be incorporated into the mosaic.

The drift-compensated relation of (4.5) will be used to compute the motion from the current camera position to the position at the first frame using the procedure detailed in Chapter 3. Thus, the drift in the estimated motion will be decreased if the UAV flies over areas that were previously registered, and it will avoid a continuous growth of the error.

4.4 Mosaic updating

Finally each pixel of frame i is transformed using the computed homography and inserted into the mosaic. The values of the pixels are interpolated using the closest pixel approximation. Other interpolation techniques like bilinear or trilinear were tested but worse results were obtained because these algorithms do not take into account the existing information in the mosaic.

Special care is necessary in the image insertion process in order to increase the mosaic coherence. In outdoors scenarios, the illumination of the images usually changes drastically, mainly due to the camera auto-exposure function. Thus, it is necessary to equalize the illumination of the images introduced in the mosaic. Once the image is warped, it is possible to compare the pixel values between the mosaic $\mathbf{M}_{i-1}$ and



(a) (b)

Figure 4.4: Illumination correction. (a) segment of mosaic without illumination correction. (b) segment of mosaic with illumination correction

the image $\mathbf{I}_i^w$. Given the set of overlapped pixels ζ_i and the size of the set $|\zeta_\mathrm{i}|$, the illumination correction factor for the image i is defined as:

$$\mu_i = \frac{\sum_x (M_{i-1}(x) - I_i^w(x))/2}{|\zeta_i|}, \quad \forall x \in \zeta_i \tag{4.6}$$

This factor is an estimation of the illumination error average in the overlapped area. It is important not to use saturated pixels (minimum or maximum gray levels) in the computation of μ_i in order to increase the accuracy of the average. This value μ_i is added to the pixel values of the warped image I_i^w, which is finally introduced in the mosaic. An example of the illumination correction is presented in Fig. 4.4 where a section of mosaic is shown with and without correction.

Finally, Fig. 4.5 and Fig. 4.6 show two 3300×3500 mosaics built with more that 500 images captured by the autonomous blimp KARMA, developed by LAAS (Laboratoire d'Architecture et d'Analyse des Systèmes) at Toulouse (Hygounenc et al., 2004), during the COMETS project general experiments (Ollero et al., 2005). KARMA flew at approximately 150m of altitude in these experiments. The mosaic computation of Fig. 4.6 involved all the techniques above described: local alignment, global alignment error reduction and illumination error correction, while in the mosaic of Fig. 4.5 no corrections were applied. The differences on the illumination can be seen.

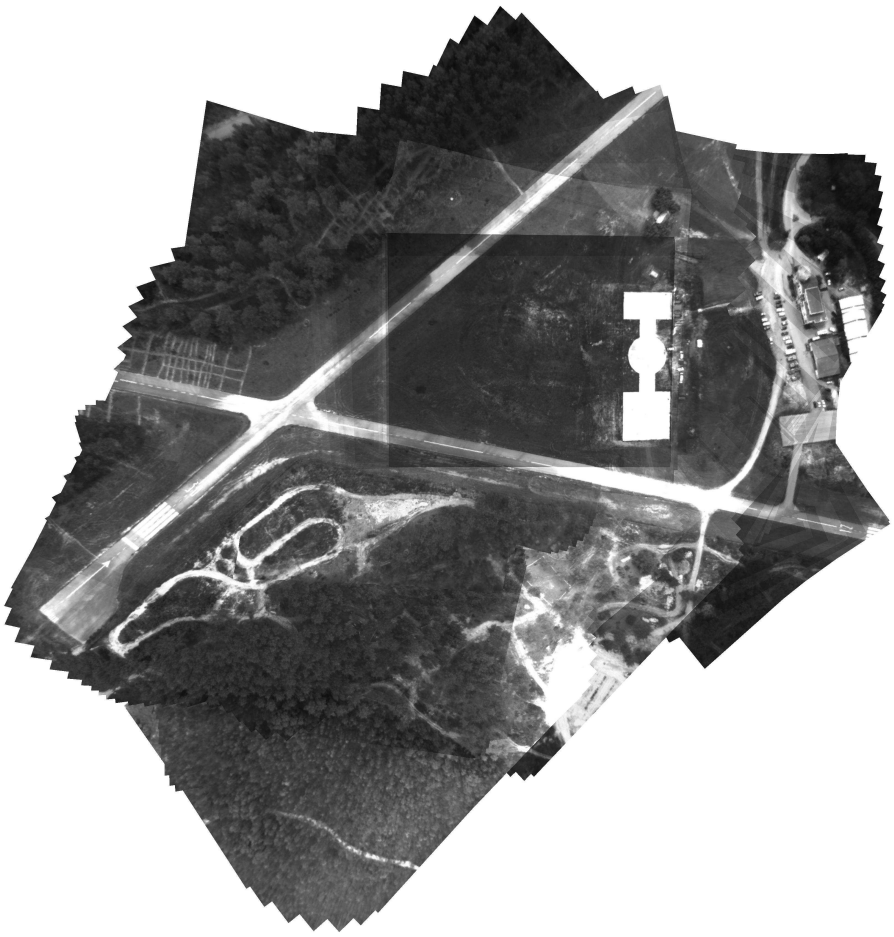


Figure 4.5: Lousa Airfield (Portugal). 3300×3500 mosaic without global alignment correction

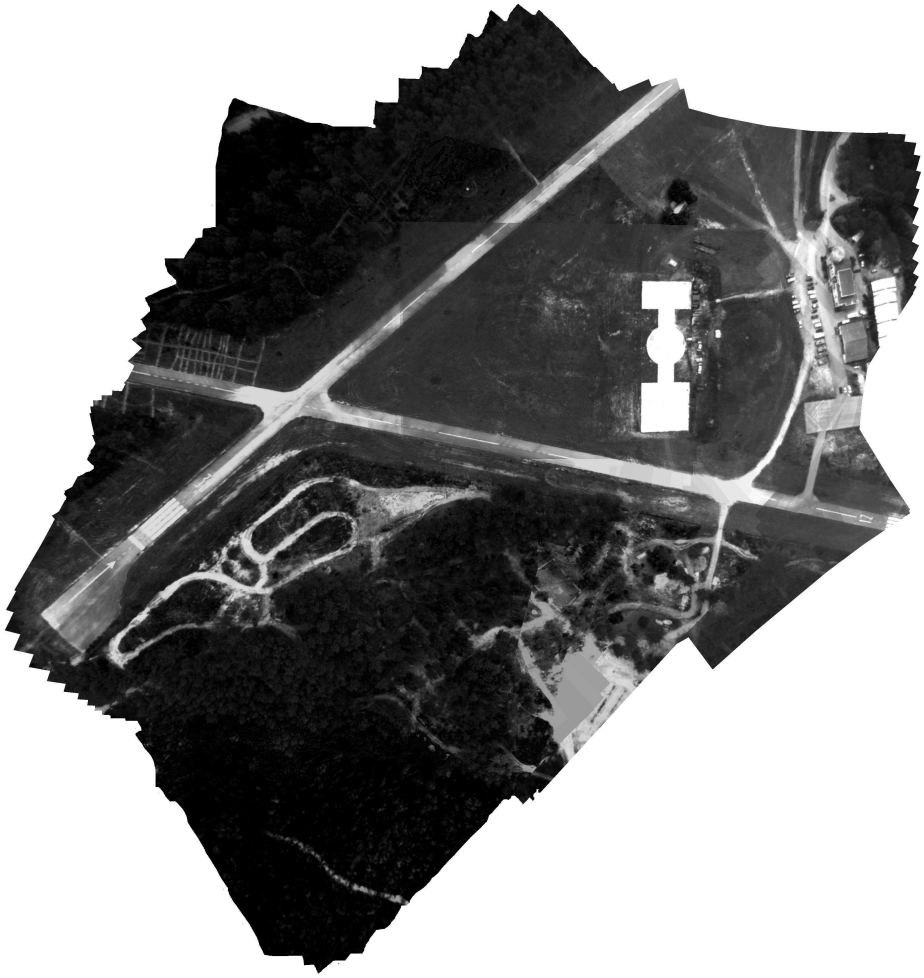


Figure 4.6: Lousa Airfield (Portugal). 3300×3500 mosaic composed by more than 500 aerial images, with global alignment and color correction

4.5 Results on motion estimation refinement

The above techniques have been tested using the autonomous blimp KARMA. The images gathered in these tests where used to build the mosaic of Fig. 4.6.

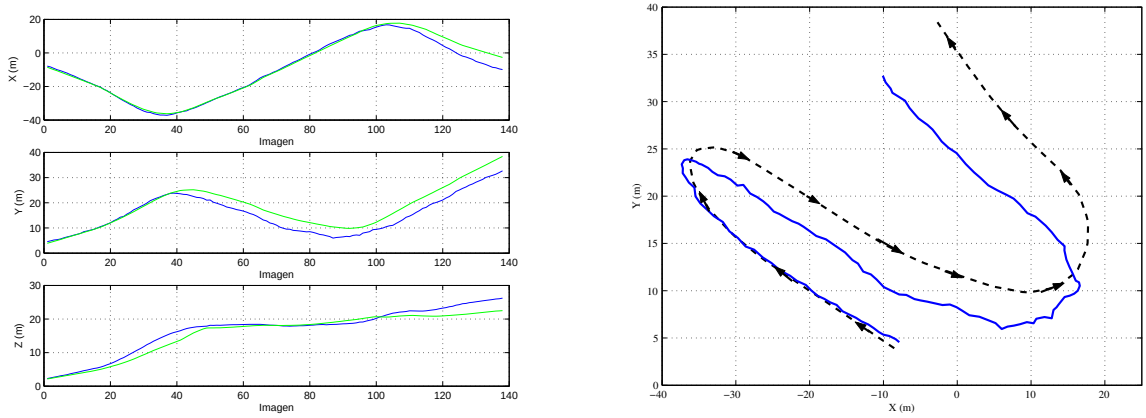
The homography-based odometry algorithm has been applied to these images, and the results compared to those obtained with the same algorithm improved with the mosaic-based global error correction. Besides, the sensors carried by KARMA (GPS, IMU and altimeter) were used to validate the estimated position from the proposed algorithms.

The experiments were carried out in an AMD Athlon XP 2200+ computer with 1GByte of RAM memory. The algorithm ran at 1 fps for full resolution images (1024x768).

Figure 4.7 shows the results of motion estimation without mosaic-based correction versus the GPS measurement. In this experiment the first 140 images employed to build the mosaic of Fig. 4.6 were used. It can be seen that although the errors are initially small, they eventually drift. The accumulative registration error is particularly evident in Fig. 4.7(b).

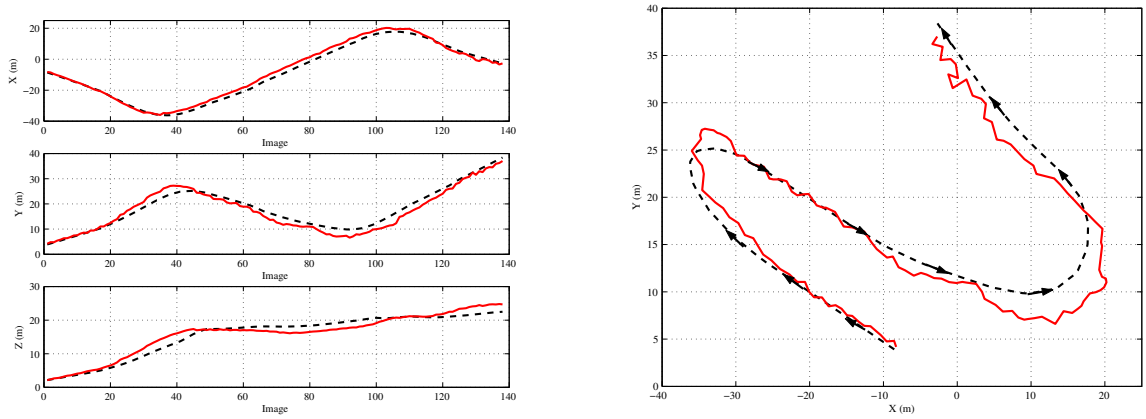
Figure 4.8(a) shows the same plots by using the mosaic to improve the position estimation. In this case it can be seen how the positioning error at the end of the trajectory is smaller than the estimated without mosaic building. The positioning error reduction can be more easily seen in Fig. 4.8(b) where the XY trajectory is plotted. Fig. 4.9 shows the 3D trajectory estimation. This raw estimation could be further smoothed with a low-pass filter in order to be used for motion control purposes.

Finally, Fig. 4.10 shows the errors on the estimated translation with respect the GPS in meters. It can be seen that, by using the mosaic correction, the error does not follow the accumulative drifting behavior associated to the single homography based motion computation.



(a) (b)

Figure 4.7: Experiments at Lousa Airfield (Portugal). The GPS translation is shown by the dashed line. The solid line is the estimated position by using the multi-view planar motion algorithm without mosaic improvement. (a): Translation in the X (Easting), Y (Northing) and Z (Altitude) axis per image (axis in meters). (b): Translation in the X-Y plane.



(a) (b)

Figure 4.8: Experiments at Lousa Airfield (Portugal). The GPS translation is shown by the dashed line. The solid line is the estimated position by using the multi-view planar motion algorithm with the mosaic-based improvement. (a): Translation in the X (Easting), Y (Northing) and Z (altitude) axis per image (axis in meters). (b): Translation in the X-Y plane.

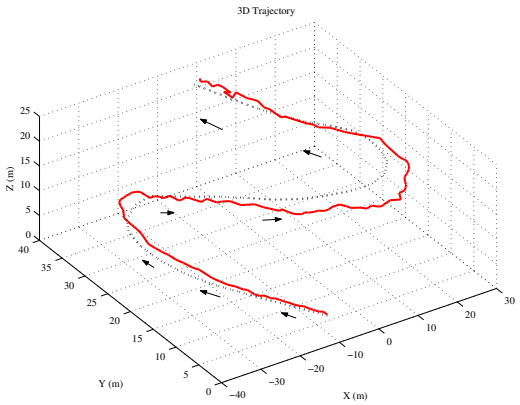


Figure 4.9: Experiments at Lousa Airfield (Portugal). 3D view of the UAV trajectory. The dashed line is the GPS translation. The solid line is the estimated position using planar motion with mosaic building improvement.

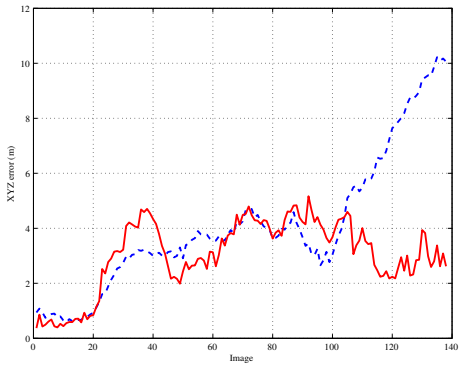


Figure 4.10: Experiments at Lousa Airfield (Portugal). Estimated position error in meters. Dashed: the error without mosaic building improvement. Solid: the error obtained with the mosaic-based correction.

4.6 Conclusions

The Chapter shows a procedure for UAV position estimation from monocular image sequences. The technique can be used for planar scenes, and requires a sensor to obtain the initial distance from the camera to the plane. The planar assumption usually holds when an UAV is flying at high altitude. The global error correction through mosaicking allows to reduce the drift on the estimation. The procedure can be implemented using low-cost and low-weight hardware, and can be a good solution for aerial vehicles.

The method can also be used with helicopters for the application of building wall inspection, in order to estimate the relative displacement of the vehicle respect to the wall as detailed in (Caballero et al., 2005), and also to generate, at the same time, high resolution images that can be used for the detection of structural damage in buildings or thermal leakage, if infrared images are used.

The mosaic can be potentially costly in terms of memory, but the amount of the needed storage capacity can be bounded by using a fixed resolution, which implies a limited level of detail for the mosaic. In practice, a limited resolution leads to restrict the applicability of the method to an interval of altitudes over the terrain; otherwise the changes of scale would not be acceptable to match the images and the mosaic. An adaptive multi-scale version of the mosaic could then be used.

However, the computed mosaic is a static picture in which the corrections cannot affect already covered areas. It means that the accuracy of the different areas of the mosaic will be fixed once the images are introduced, even if the global alignment procedure provides better positioning. If the relations among the images are flexible, then the corrections could be introduced and the mosaic improved.

Chapter 5

A stochastic framework for mosaic building

This Chapter presents a probabilistic framework where uncertainties can be considered in the mosaic building process. The homography combined with its covariance matrix are used to link the images of the mosaic. Moreover, the Chapter describes how, when a loop is present in the sequence of images, the accumulated drift can be compensated and propagated to the rest of the mosaic. The proposed method improves the mosaic building process and hence the mosaic-based localization described in Chapter 4.

5.1 Introduction

Knowledge about the environment is a critical issue for autonomous operations. In general, the environment representation depends on the kind of sensors used to estimate it and the tasks and circumstances in which the vehicle will be involved. This Chapter addresses the environment representation problem for aerial vehicles and its estimation by means of monocular imagery.

In the case of an aerial robot that is not affected by obstacles at the flight altitude, geo-referenced mosaics can be sufficient as environment model for certain tasks. A

mosaic is built by aligning to a common frame a set of images gathered while the aerial vehicles is moving.

If the scene is planar (which is, in general, a valid approximation when the ratio between the distance to the scene and the ground elevation is high), a set of matches between two views can be used to estimate a homographic model for the apparent image motion. This model can be composed with the estimated motion of the rest of images in order to build the mosaic. Moreover, Chapter 4 showed that the motion of the aerial vehicle can be derived from the alignment of its images with the mosaic for a planar scene.

However, the approach contains small image motion inaccuracies that lead to an erroneous estimation of the mosaic and, as a result, to a drift in the estimated position of the robot when large areas are covered. This problem is solved in (Garcia et al., 2002) by means an online mosaicking architecture that allows to improve the alignment of the central pixel of the images within the mosaic when loops are detected. The technique works well but it is a waste of information because, in our case, there exist a full covariance matrix of the homography that can be used to propagate the error information through the mosaic.

This Chapter proposes a new technique in which a complete homography and its associated full covariance matrix are used to represent the alignment of the images in the mosaic. It draws an accurate model that considers the uncertainties in the local relations between the images that compose the mosaic. The proposed mosaic architecture allows the detection of loops and hence the reduction of the image drift which is present in this kind of methods. Furthermore, the uncertainty reduction in the local image alignment can be propagated to the rest of the mosaic thanks to the full covariance matrix computation.

The Chapter describes the steps needed to introduce uncertainties in the mosaic building procedure by means of the Kalman filter. Firstly, an overview of the approach is presented. Then, the estimation of the local relations among images is outlined. Finally, the proposed Kalman filter for stochastic mosaicking is detailed and some experimental results are shown.

5.2 Technique overview

The motivation of this work are the results presented in Chapter 4, where monocular imagery is used to compute the real motion that a camera attached to an aerial robot undergoes. The odometer presented there is based on the fact that it is possible to obtain the motion of a calibrated camera, up to a scale factor, from the homographic models that relate several images of a planar scene. The local homographies between consecutive images can be composed to obtain the global motion of the vehicle, but the local errors lead to a progressive drift in the estimated location of the vehicle. Furthermore, the mosaic can be used to decrease the accumulative errors associated to the odometry. However, the construction of the mosaic itself is affected by errors, which are not considered.

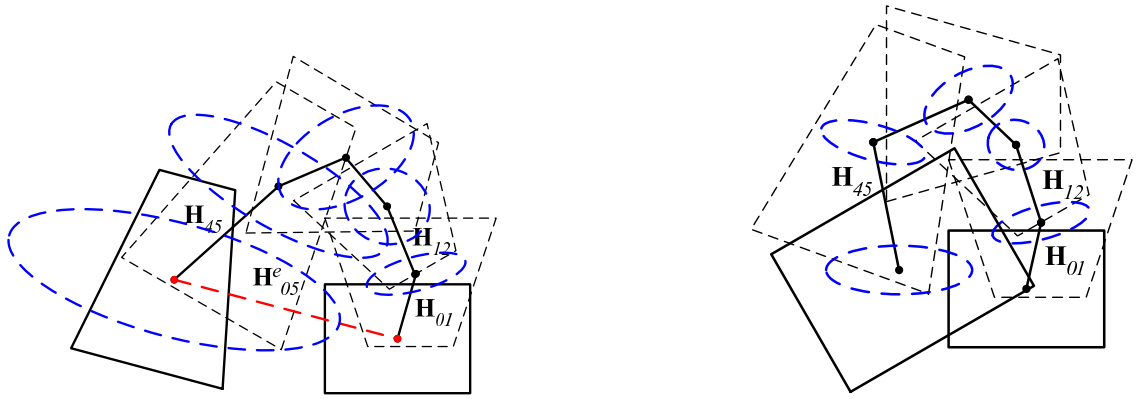
In the approach proposed in this Chapter, the mosaic building technique is improved at two levels:

- A new approach is presented for the homography computation in pseudo-planar scenes where the planar assumption may not hold. A homography model relaxation is imposed in order to guarantee the computation. The covariance matrix associated to the homography will store the accuracy information and then be transferred to the mosaic.

- The covariances of the estimated homographies are used in a new mosaic building architecture. When a zone of the mosaic is revisited, a procedure is used to decrease the accumulated drift. A full covariance matrix that considers the correlation among the estimated homographies in the loop is computed, and the stochastic information is employed to propagate the correction to all the images involved (see Fig. 5.1).

5.3 Hierarchical homography computation

The technique proposed to build the mosaics is based on the one presented in Chapter 4. A careful analysis points out two factors that may reduce the applicability of the



(a) (b)

Figure 5.1: Basic procedure: (a) when a part of the mosaic is revisited, (b) the drift is corrected and the correction propagated. Although only the covariances on the position are shown (which are used to determine potential loops), the full covariance matrix for all homographies is maintained.

technique, mainly when the UAV flies at altitudes of the same order of other elements on the ground (buildings, trees, etc):

- In 3D scenes, the parallax effect will increase, and the planarity assumption may not hold. The result is a dramatic growth of the number of outliers and even the divergence of the M-Estimator.

- Depending on the frame-rate and the vehicle motion, the overlap between images in the sequence is sometimes reduced. This generates a non-uniform distribution of the features along the images.

The goal of this section will be to increase the reliability of the homography computation in order to face flights at medium/low altitude.

The above problems are related, in the sense that both generate an ill-posed system of equations for the computation of the homography. If the matches are not uniformly distributed over the images, there may exist multiple solutions; and if the parallax effect is significant, there may exist multiple planes (whose transformation should be described by multiple homographies).

A classical solution to improve the results is to introduce additional constraints to reduce the number of degrees of freedom of the system of equations. In the proposed solution, this is accomplished through a hierarchy of homographic models (see Fig. 5.2), in which the complexity of the model to be fitted is decreased whenever the system of equations is ill-constrained.

Therefore, depending on the quality of the available data, the constraints used to compute the homography are different. An estimation of this accuracy will be given by the covariance matrix of the computed parameters. If this stochastic information is available, it can be used to define the accuracy of the image alignment inside the mosaic.

5.3.1 Implementation

The following paragraphs presents the way to progressively reduce the complexity of the homographic model and the laws that govern the transitions among the hierarchy.

From Chapter 2, it can be seen that any non-singular invertible $3x3$ matrix can be considered as a homography:

$$\mathbf{H} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \tag{5.1}$$

A complete homography has 8 degrees of freedom (as it is defined up to a scale factor). The degrees of freedom can be decreased by fixing some of the parameters of the $3x3$ matrix. The models used are the defined by Hartley in (Hartley and Zisserman, 2004): Euclidean, Affine and Complete Homographic models, which have 4, 6 and 8 degrees of freedom respectively (see figure 5.2).

The percentage of successful matches obtained by the point tracker is used to have an estimation about the level of the hierarchy where the homography computation should start. These percentage thresholds were obtained empirically by processing hundreds of aerial images. Each level involves the following different steps:

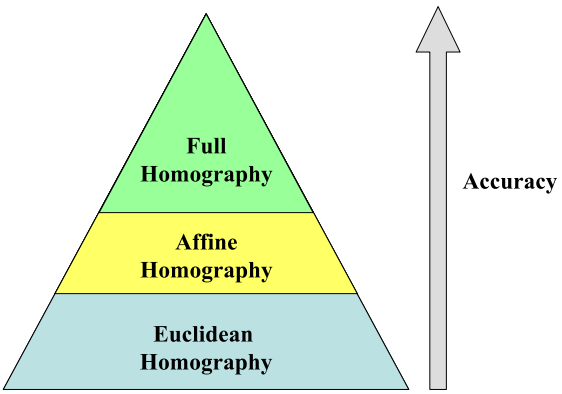


Figure 5.2: Levels in the proposed hierarchical homography computation. Accuracy increases with the complexity of the model.

- Complete homography. This model is used if more than the 65% of the matches are successfully tracked. Least median of squares (LMedS) is used for outlier rejection and a M-Estimator to compute the final result. It is the method proposed in Chapter 2 to compute the homography.

- Affine homography. If the percentage of success in the tracking step is between 40% and 65%, LMedS is not used, given the reduction in the number of matches. A relaxed M-Estimator (soft penalization) is carried out to compute the model.

- Euclidean homography. If the percentage is below 40%, the set of data is too noisy and small to apply non-linear minimizations. The model is computed by using just least-squares.

In addition, it is necessary a rule to know when the current level is not constrained enough and the algorithm has to decrease the model complexity. The M-Estimator used in the complete and affine computations is used for this purpose. Thus, if it diverges, the level in the hierarchy has to be changed to the next one. It is considered that the M-Estimator diverges if it reaches the maximum number of iterations.

5.3.2 Experimental results

This section shows some results of the proposed technique in real conditions. The algorithm was applied to a long image sequence captured at relatively low altitude, and the computed homographies were used to align each new image with a mosaic. This experiment tries to visually demonstrate the coherence of the homographies computed by using the proposed technique but not to build an accurate mosaic.

Figure 5.3 shows the mosaic built with more than 400 images taken by KARMA. The scenario is a parking at LAAS, and the distance from the blimp to the ground is 27 meters. The low altitude makes evident the parallax effect; some trees are 15 meters high.

The technique described in Chapter 4 was applied to this image sequence but the parallax effect made impossible the homography computation during the processing and only a small portion of the mosaic was computed. Figure 5.4 shows that the scene is not planar and confirms the feasibility of the algorithm. On the other hand, the mosaic could be computed by using the hierarchical computation described in this Chapter.

The image sequence used to build the mosaic (figure 5.3) was processed at a rate of 2 images per second with resolution of 1024x768. The final mosaic has a 4300x2400 effective resolution.

5.4 Stochastic mosaicking

The goal of this section is to improve the environment model by attaching stochastic information to the mosaic. This information will be used to improve the positions of the images within the mosaic when a close-loop is detected. In this approach, the mosaic is represented by a database of images and associated data; the mosaic is no longer a static image, but a set of images linked by stochastic relations.

Thus, when a close-loop is detected the relations among the images involved in the loop are updated according to the accuracy of the measurement and the estimation

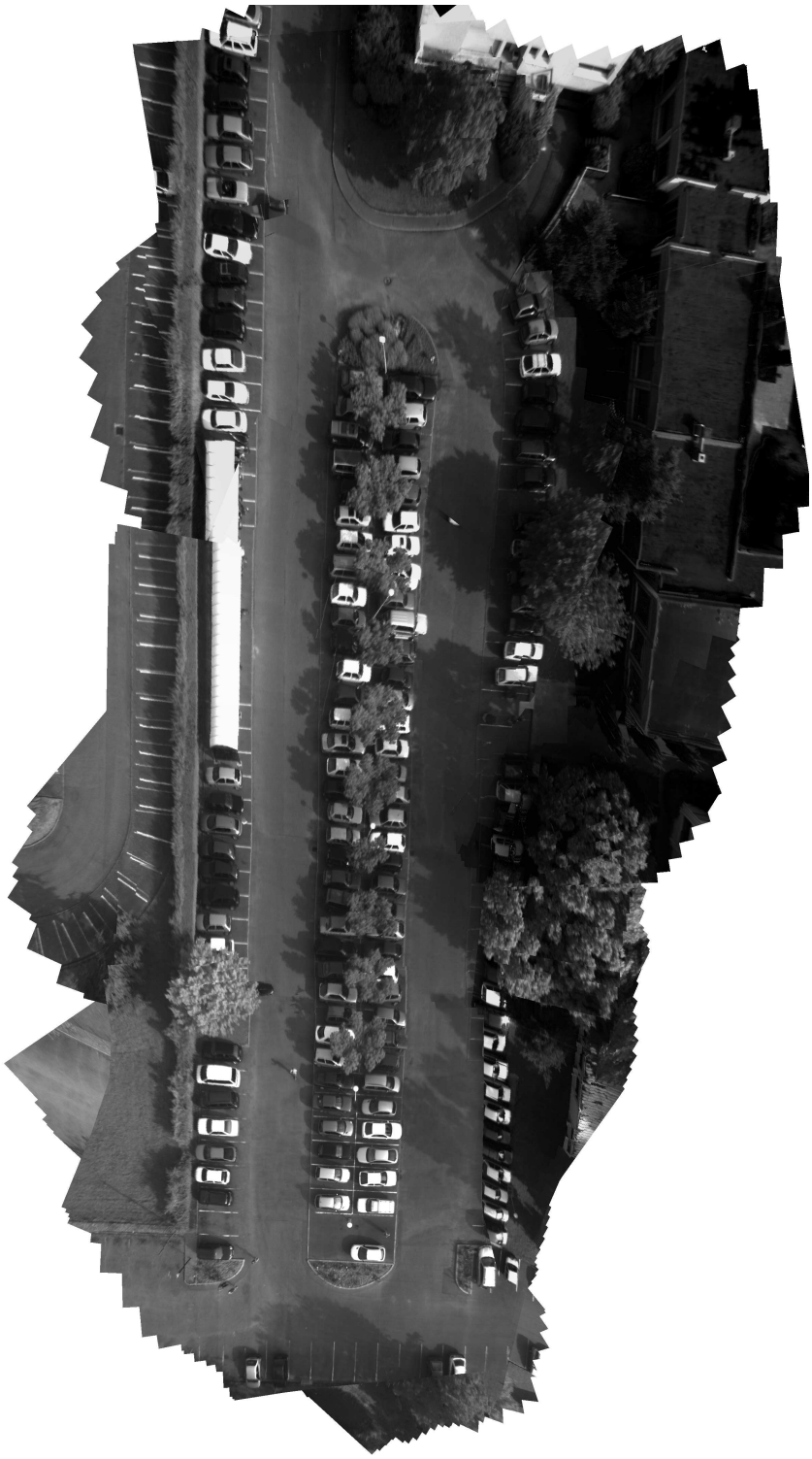


Figure 5.3: Mosaic of "le petit parking" at LAAS (Toulouse). Images taken at 27 meter over ground. Images taken by KARMA

(a) (b)

Figure 5.4: Two images extracted from the sequence used to build the mosaic of figure 5.3. It is clear that not all the features can be considered to lay in the same plane.

of the position of each image. A minimization process based on a Kalman Filter is used to optimize the estimation of the inter-image relations.

It is well known that the Kalman Filter (Kalman, 1960) is a set of mathematical equations that provides efficient computational means to estimate the state of a process, in a way that minimizes the mean of the squared error, when the system dynamics are linear and existing noise follows a Gaussian distribution. The filter is very powerful in several aspects: it support estimation of past, present, and even future states, and it can do so even when the precise nature of the modelled system is unknown.

The filter allows to analyze time-varying physical systems in the presence of noise. The system is modelled by a state vector $\mathbf{x}$ that has entries for each of the relevant variables. The knowledge about the expected state is stored in the state transition function $\mathbf{f}$.

The filter allows a continuous and efficient estimation of $\mathbf{x}$, incorporating the information provided by any measurement $\mathbf{z}$ composed by a set of accesible variables whose dependence on the state $\mathbf{x}$ is known. The current state estimation is stored in the vector $\hat{\mathbf{x}}$ and its uncertainties are encapsulated in the covariance matrix $\mathbf{P}$. If the dependence of both $\mathbf{f}$ and measurement function $\mathbf{h}$ on $\mathbf{x}$ is linear, and the statistical distribution of noise is Gaussian, then the estimation of $\mathbf{x}$ by the filter is optimal.

Given the linear system:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}(k-1) + \mathbf{w}(k-1) \quad (5.2)$$

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k-1) + \mathbf{n}(k) \quad (5.3)$$

The Kalman filter proposes the following equations to predict the new state and its covariance matrix:

$$\hat{\mathbf{x}}^{-}(k) = \mathbf{A}\hat{\mathbf{x}}(k-1) + \mathbf{B}\mathbf{u}(k-1) \quad (5.4)$$

$$\hat{\mathbf{P}}^{-}(k) = \mathbf{A}\mathbf{P}(k-1)\mathbf{A}^{\mathbf{T}} + \mathbf{Q} \quad (5.5)$$

The new measurement $\mathbf{z}$ is considered by means of:

$$\mathbf{K}(k) = \mathbf{P}^{-}(k)\mathbf{H}^{\mathbf{T}}(\mathbf{H}\mathbf{P}^{-}(k)\mathbf{H}^{\mathbf{T}} + \mathbf{R})^{-1} \quad (5.6)$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^{-}(k) + \mathbf{K}(k)(\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}^{-}(k)) \quad (5.7)$$

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H})\mathbf{P}^{-}(k) \quad (5.8)$$

Where $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices associated to the process noise and measurement noise respectively.

The Extended Kalman Filter makes use of the linearized system of equations to provide the same functionality when the system dynamics are not linear. More information and further details on Kalman filtering can be found in (Welch and Bishop, 1995).

5.4.1 Closing the loop

Each time a new image is gathered by the cameras of the UAV, the homography that relates it with the previous one is computed, as well as its covariance. The position of the image inside the mosaic is obtained by multiplying the current homography by all the previous homographies until the reference frame is reached, as presented in Chapter 4. Then, the image is introduced in the database with the following information:

homography with respect to the previous image, composed homography to reference frame, covariance matrix of each homography and other positioning information.

The loop-closing is computed by detecting the crossover of the current image with one or more images of the database. For this purpose, the estimated position (and its covariance) of the central pixel of each image in the mosaic is stored in the database. The crossover detection consists of finding one image whose Mahalanobis distance (Mahalanobis, 1936) to its central pixel is within a certain empirical range.

Once the crossover is detected, a feature matching procedure is launched to compute the alignment between both images. In the general case, the task of matching images taken from very different points of view is difficult and computationally costly. Even if a good estimation of the translation between images is available, the tracking algorithm (Ollero et al., 2004) cannot deal with a rotation higher that 45 degrees, as the features used are not affine invariant.

If a pseudo-planar scene is considered, the matching complexity can be drastically reduced. An initial estimation of the location (a complete homography) of the image inside the mosaic is available, so this estimation can be used as a searching seed for the feature matching procedure. At time i the homography that aligns the current image $\mathbf{I}_i$ to the mosaic is $\hat{\mathbf{H}}_{0i}$. If a crossover with image $\mathbf{I}_j$, whose estimated position inside the mosaic is given by $\hat{\mathbf{H}}_{0j}$, is detected, the initial estimation of the transformation from $\mathbf{I}_i$ to $\mathbf{I}_j$ is defined by:

$$\hat{\mathbf{H}}_{ji} = \prod_{k=j+1}^{i} \hat{\mathbf{H}}_{(k-1)k} \tag{5.9}$$

If the estimation error for each homography were zero, the alignment of $\mathbf{I}_i^w$ (the result of warping $\mathbf{I}_i$ according to $\hat{\mathbf{H}}_{ji}$) with respect to $\mathbf{I}_j$ would be perfect, but inaccuracies in the estimated homographies lead to unavoidable alignment errors.

However, computing matches between the warped image $\mathbf{I}_i^w$ and $\mathbf{I}_j$ is an easier task. From this data it is possible to obtain the homography $\mathbf{H}_{ji}^e$ that describes the drift between both images and then obtain the correct alignment:

$$\mathbf{H}_{ji} = \hat{\mathbf{H}}_{ji}\mathbf{H}_{ji}^e \tag{5.10}$$

It is clear that, by using this method, the matching algorithm does not have to deal with the complete motion between both images, but only with the accumulated errors. Then, it will reduce the complexity of the matching algorithm needed and will improve the alignment results.

In addition, using normalized correlation allows to deal with the illumination problems that appear in close loops when images are taken at different time of day. Erroneous matches associated to different causes like shadows and parallax are later removed by the outlier detection and robust estimation performed in the homography computation.

5.4.2 Updating the mosaic after a loop-closing

As stated before, when a loop-closing is detected, a measurement about the alignment between the current image $\mathbf{I}_i$ and the crossover $\mathbf{I}_j$ is given by the system. This measure is a homography $\mathbf{H}_{ji}$ and its covariance matrix $\mathbf{C}_{\mathbf{h}_{ji}}$. The problem now is how to update the relations among the images from $\mathbf{I}_j$ to $\mathbf{I}_i$ under the following constraint:

$$\mathbf{H}_{ji} = \prod_{k=j+1}^{i} \hat{\mathbf{H}}_{(k-1)k} \tag{5.11}$$

For this purpose, a minimization process based on the Extended Kalman Filter is launched each time a loop-closing is detected. This filter re-estimates the relations among the images involved in the loop, taking into account the uncertainty of the loop-closing and the stochastic information stored in the database. This procedure is summarized if Fig. 5.5. Next sections will detail the structure and dynamics of the filter.

Notation

For this section, the following vector will be considered as the vector representation of the homography matrix $\mathbf{H}$:

$$\mathbf{h} = \begin{bmatrix} h_{00} & h_{01} & \dots & h_{22} \end{bmatrix}^T \tag{5.12}$$

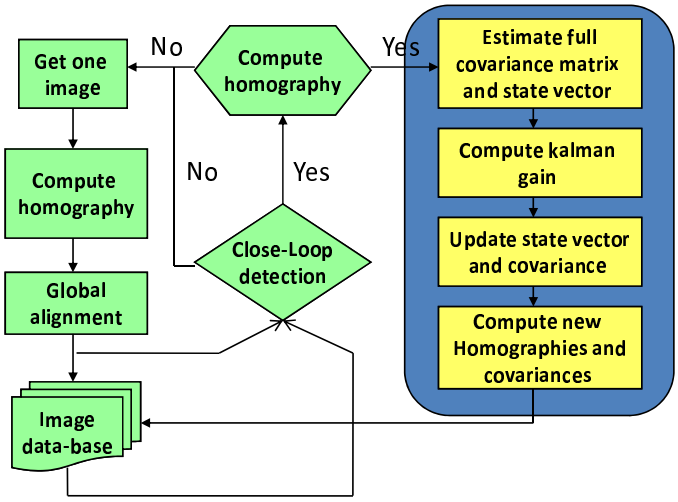


Figure 5.5: Stochastic mosaicking schema

The $\mathbf{C_h}$ covariance matrix is then represented by a 9×9 matrix with the variance of these variables in the diagonal and the cross-variances in the upper and lower triangles.

In the following, the symbol "$\cdot$" stands for the product of two homographic vectors which result is expressed as a vector too. Thus, the operation $\mathbf{h}_1 \cdot \mathbf{h}_2$ is equivalent to express as vector the result of the algebraic product $\mathbf{H}_1\mathbf{H}_2$

The state vector

It is assumed that a close-loop is detected between $\mathbf{I}_j$ and $\mathbf{I}_i$, with $n+1$ images involved, and the transformation that aligns both crossover images is $\mathbf{H}_{ji}$ with covariance $\mathbf{C}_{\mathbf{h}_{ji}}$. For simplicity, it can be assumed that $j = 1$ and $i = n+1$.

The a priori state vector will be composed by the n transformations that align the n images $(\mathbf{I}_2, ..., \mathbf{I}_{n+1})$ with $\mathbf{I}_1$, thus:

$$\mathbf{x}^- = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T = [\mathbf{h}_{12}, \mathbf{x}_1 \cdot \mathbf{h}_{23}, ..., \mathbf{x}_{n-1} \cdot \mathbf{h}_{n(n+1)}]^T \tag{5.13}$$

where $\mathbf{x}_i$ and $\mathbf{h}_{ij}$ are the vector representation of the homography matrices $\mathbf{X}_i$ and $\mathbf{H}_{ij}$, as shown in equation (5.12).

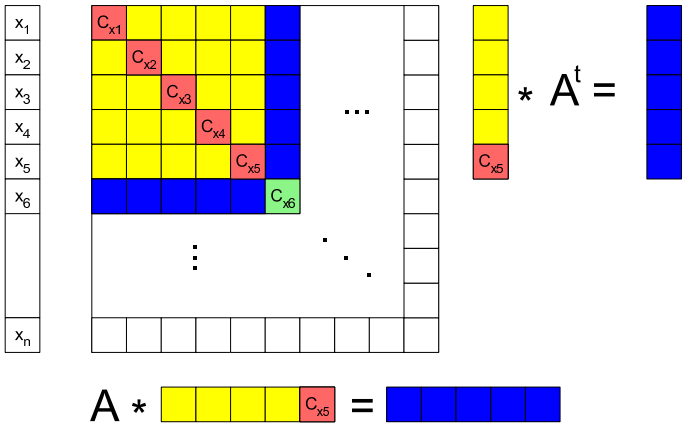


Figure 5.6: Generation of the a priori covariance matrix of the state vector

It is easy to see that the state (5.13) is obtained by recursively applying equation (5.9): $\mathbf{x}_1$ is the product at $k = j + 1 = 2$, $\mathbf{x}_2$ is the product at $k = j + 2 = 3$ that can be written in function of $\mathbf{x}_1$ and so on.

Prediction stage

The calculation of the a priori state vector and its a priori covariance matrix $\mathbf{P}^-$ is done in an incremental fashion for the n states using the following prediction equation:

$$\mathbf{x}_i = \mathbf{x}_{i-1} \cdot \mathbf{h}_{i(i+1)} \tag{5.14}$$

It is easy to compute the Jacobian of this expression with respect to the state vector (matrix $\mathbf{A}$) and with respect to the variables of $\mathbf{h}_{i(i+1)}$ (matrix $\mathbf{W}$). $\mathbf{A}$ will be used to generate the corresponding rows and columns of the matrix $\mathbf{P}^-$ (see figure 5.6) and, with $\mathbf{W}$, to compute the covariance matrix of $\mathbf{x}_i$ by means of:

$$\mathbf{C}_{\mathbf{x}_i} = \mathbf{A}\mathbf{C}_{\mathbf{x}_{i-1}}\mathbf{A}^T + \mathbf{W}\mathbf{C}_{\mathbf{h}_{i(i+1)}}\mathbf{W}^T \tag{5.15}$$

The structure of the covariance matrix shows how every homography is fully correlated to the previous ones in the loop.

Updating stage

The previously described state vector is arranged in order to minimize the number of operations needed to update it with the measurement ($\mathbf{H}_{1(n+1)}$,$\mathbf{C}_{\mathbf{h}_{1(n+1)}}$). Thus, the state $\mathbf{x}_n$ of $\mathbf{x}^-$ represents the transformation from the current image $n+1$ to the loop reference frame 1, so $\mathbf{h}_{1(n+1)} = \mathbf{x}_n$. This fact is described in the following measurement equation:

$$\mathbf{h}_{1(n+1)} = \mathbf{F}\mathbf{x} = \left[\mathbf{0}_{9x(9n)}, \mathbf{I}_{9x9}\right] \mathbf{x} \tag{5.16}$$

It is easy to update the state vector and its covariance matrix following the classic Extended Kalman Filter equations. This simple measurement function $\mathbf{F}$ will significantly reduce the computation needed to obtain the gain $\mathbf{K}$ and, later, the a posteriori covariance matrix $\mathbf{P}$.

Particular attention is required for the state vector updating:

$$\mathbf{x} = \mathbf{x}^- + \mathbf{K}(\mathbf{h}_{1(n+1)} - \mathbf{F}\mathbf{x}^-) \tag{5.17}$$

The homographies are defined up to scale factor; thus, a homography multiplied by a constant k represents exactly the same transformation, although its components are different to those of the original matrix. To implement the substraction of homographies in (5.17) it is necessary to perform a normalization of the scale factor. The proposed solution is to set the determinant of both homographies to 1 before computing the state vector.

Finally, the relations among the images involved in the loop are updated with the measurement but they are expressed in the local coordinate frame of the loop (the first image of the loop, $\mathbf{I}_1$, is the reference frame) and have to be transformed into the mosaic system reference. From equation (5.14):

$$\mathbf{H}_{i(i+1)} = (\mathbf{X}_{i-1})^{-1}\mathbf{X}_i \tag{5.18}$$

and the covariances can be derived from the Jacobian of this expression.

5.5 Results of vision-based localization experiments

This section shows some experiments to demonstrate the correct operation of the proposed algorithm. The framework of the experiments is the position computation by means of monocular imagery. The position of the vehicle is estimated by means of the algorithm described in Chapter 3 and the inputs are the homographies computed with the proposed stochastic mosaicking.

The images were taken by KARMA during one experiment in the parking at LAAS. The sequence is composed by two hundred images. Figure 5.7 shows the mosaic built with the images and the computed homographies.

More that 15 short loops (less than 20 images involved) and one large loop (more than 100 images) were closed during the experiment. In figure 5.8 the image based X, Y, Z and XY position estimation and GPS position measurement are displayed. Notice the sudden left turn in the vision based estimation at coordinates (31, -50) in the XY estimation, it is automatically carried out when the plane induced by the floor (the ground truth) becomes the scene dominant plane again.

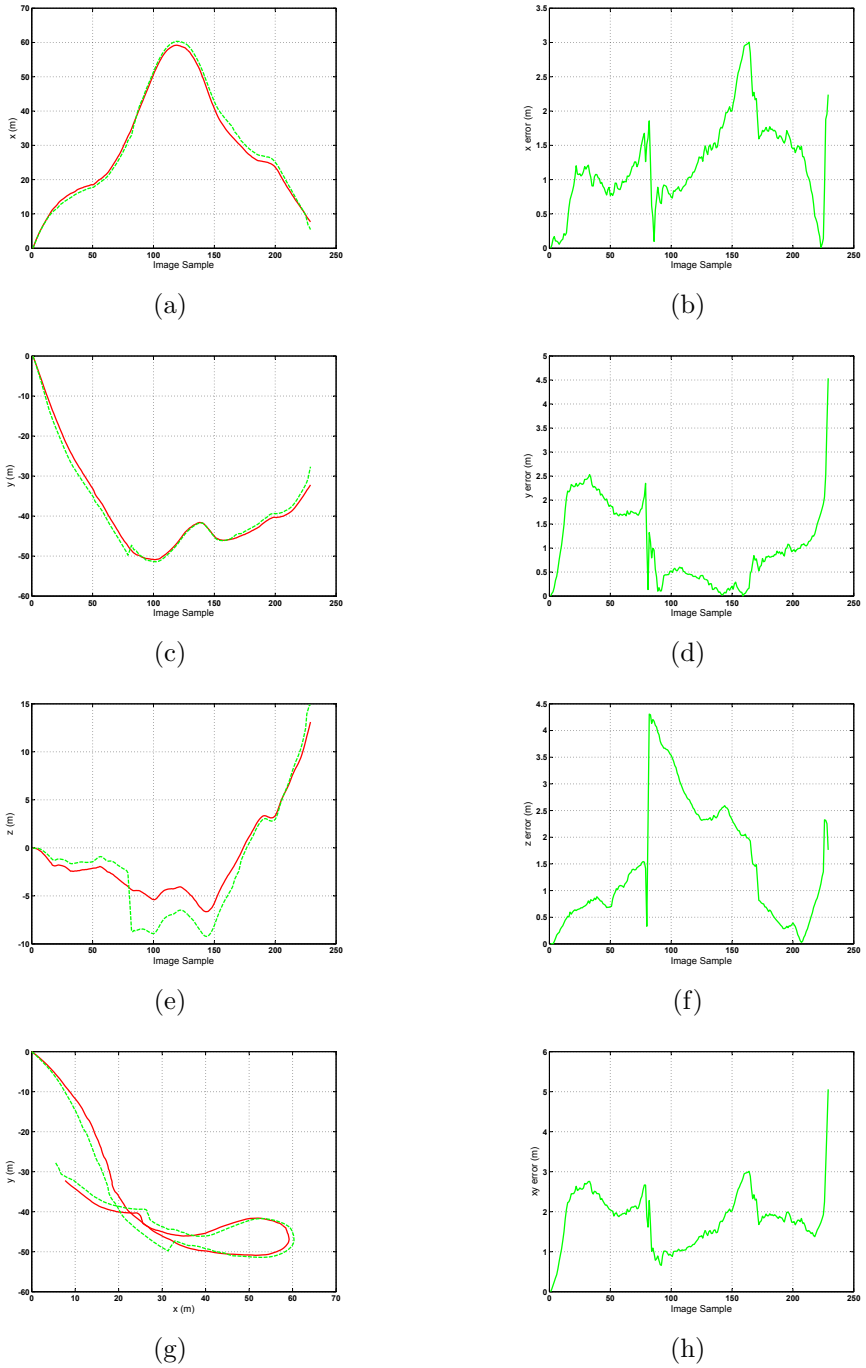
Figure 5.8 also shows the evolution of the errors associated to the position estimation. The error is calculated as the Euclidean distance between the GPS and the estimated position at each image sample. It can be seen that the error is moderate: the maximum value in the XY plane is 3 meters and the mean is 1.76 meters.

5.6 Conclusions

Mosaics can be a convenient environment representation for UAV tasks, such as monitoring events, identification of changes and others. The Chapter presents an approach for building mosaics based on planar homographies. A hierarchical homography computation increases the robustness of the approach, allowing to deal with quasi-planar scenes.

An important aspect of the Chapter is the inclusion of uncertainty measures in the mosaic. The mosaic is no longer a static picture, but a collection of relations among images. Although the alignment errors will grow over time, they can be partially reset

Figure 5.7: Mosaic of “le grand parking” at LAAS (Toulouse). KARMA was flying at 22 meter over the ground.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.8: Position estimation experiment with mosaic computation. Left: Image based position estimation (dashed line) and GPS position measurement (solid line). Right: Image based position estimation error with respect the GPS measurement

when part of the mosaic is revisited. Moreover, the correction can be propagated to the rest of the mosaic by considering the relations among images.

UAV position estimation is an interesting potential application of the proposed approach. It is possible to determine the motion of a calibrated camera from plane-induced homographies by means of the method proposed in Chapter 3. Furthermore, the proposed technique allows to reduce the accumulative drift in the position estimation when the UAV flies over previously visited areas.

The main drawback of the approach are the storage requirements. Not only the images of the mosaic should be stored (although this number can be controlled, depending on the degree of overlap), but also the state augments along the sequence of images. The current state of the technology allows to solve the image storage by means of solid state hard disks which support high capacity at really low weight. However, this approach cannot deal in real-time with the computation needed to update the mosaic in very large loops. In this sense, new techniques based on the downsampling of the homographic relations between images and the use of Sequential Map Joining (Tardos et al., 2002) will be researched.

Chapter 6

Application of Homography-based odometry to the SLAM problem

This Chapter extends localization techniques to a more general problem by applying SLAM (Simultaneous Localization And Mapping) techniques. This allows to reduce the inherent constraints associated to the method presented in Chapter 5, particularly in altitude and camera orientation.

6.1 Introduction and related work

Chapters 4 and 5 consider that the UAV flies in such a way that a mosaic can be computed and then used to refine localization. The main constraints of this method are associated to the mosaic building process. Sometimes, mosaics cannot be computed due to oblique camera orientations or significant variations in the altitude of the UAV.

This Chapter proposes a SLAM-based technique to generalize the problem of localization. The Chapter addresses the situations in which a mosaic cannot be built and another localization technique is required. The approach consists of applying the Homography-based odometry as prediction stage of a SLAM algorithm.

Simultaneous localization and mapping (SLAM) is a technique used to build a map within an unknown environment while at the same time estimating the current

position and pose of the robot. It is mainly characterized by the use of a stochastic framework to represent the relationship among the map and the position of the robot. Early researches in SLAM were presented in (Smith and Cheeseman, 1987), where the use of indirect and noisy measures to estimate the robot pose and its uncertainty is discussed.

First SLAM implementations were based on Extended Kalman filtering (EKF) and only one type of sensor, (Leonard and Durrant-Whyte, 1991; Betge-Brezetz et al., 1996; Betke and Gurvits, 1997; Feder et al., 1999). The iterative nature of the EKF allows on-line estimation of both localization and mapping in a structured manner; at the same time, it permits easy incorporation of noise, expressed as covariance matrixes. However, the EKF linearizations, that can lead to inconsistent estimations when the system equations are strongly non-linear, combined with the need of faster SLAM implementations, pushed the robotics community to research on new techniques to solve the SLAM problem. Thus, the use of Particle Filtering for the factorization of the posterior function in the FastSLAM (Montemerlo et al., 2002) and FastSlam2.0 (Montemerlo et al., 2003) allowed significant improvements in the computation. The Information Filter (the Kalman Filter dual) has recently received more attention due to its applicability in feature-based SLAM (Thrun et al., 2004). In this form, map posteriors are dominated by a small number of links that tie together nearby features in the map. This characteristic can be exploited to significantly decrease the computation load of the update equations.

This Chapter considers a particular case of SLAM problem, called bearing-only SLAM or boSLAM in which bearing only sensors are used, a camera in this case. As stated in (Vidal-Calleja et al., 2007), the boSLAM is a partially observable problem, so the depth of the landmarks cannot be directly estimated. This entails a difficult landmark initialization problem which has been tackled with two basic approaches: delayed and undelayed initialization. In delayed initialization, landmarks are not included in the SLAM system in the first observation, but after that the angular baseline in between has grown large enough to ensure a good triangulation. This method has the advantage of using well conditioned landmarks, but the SLAM system cannot take advantage of the landmark until its localization is well conditioned.

Several approaches have been proposed in this area such as (Davison, 2003) where a Particle Filter is used to initialize the landmark depth, or (Deans and Hebert, 2000) where non-linear optimization is used instead of mean and covariance matrix.

On the other hand, undelayed approaches introduce the landmark in the SLAM system with the first observation, but some considerations have to be taken into account due to the fact that the landmarks are usually bad conditioned in depth, and then may appear problems of SLAM filter divergence. Most existing approaches are based on multiple hypotheses, where a Gaussian Mixture is used for landmark initialization in a Kalman Filter, as in (Kwok and Dissanayake, 2004). Recent research (Montiel et al., 2006) proposes the inverse depth parametrization in a single-hypothesis approach for landmark initialization with promising results.

The technique presented in this Chapter proposes a new undelayed feature initialization that takes advantage of the scene normal plane estimation computed in the Homography-based odometry detailed in Chapter 3. Indeed, the technique cannot be considered as boSLAM because a range sensor is used, combined with the normal vector to the plane, to initialize the landmark depth. The approach is based on an EKF that simultaneously estimates the pose of the robot (six degrees of freedom) and the map. The use of the estimated rotation and translation provided by the odometer as the main motion hypothesis in the prediction stage of the EKF is another contribution made by this approach. Complex non-linear models are normally used to estimate vehicle dynamics, due to the lack of odometers in UAVs. This leads to poor prediction hypotheses, in terms of accuracy, and then a significant reduction of the filter efficiency. In (Kim and Sukkarieh, 2004) a solution based on merging model-based estimation and inertial measurements from local sensors (IMUs) is proposed, resulting in an accuracy growth. The integration of the IMU in the presented approach will be also considered in order to improve the position estimation.

The Chapter is organized as follows. First, a short overview of the SLAM problem is presented. Then, the full EKF-based approach is described, including landmark initialization. Finally some experimental results and conclusions are shown.

6.2 SLAM overview

The SLAM problem can be formulated as a probabilistic Markov chain process. The robot's pose at time t will be denoted as $\mathbf{p}_t$. Poses evolve according to a probabilistic law, often referred as motion model $p(\mathbf{p}_t|\mathbf{u}_t, \mathbf{p}_{t-1})$, where $\mathbf{p}_t$ is a probabilistic function of the robot control $\mathbf{u}_t$ and the previous pose $\mathbf{p}_{t-1}$.

The robot environment consists of N motionless landmarks. Each of them is characterized by its location in space, denoted as $\mathbf{y}_n$ for $n = 1, ..., N$. These landmarks will be detected by the sensors onboard the robot and used to represent the environment.

The measurement at time t will be denoted as $\mathbf{z}_t$ and will provide a set of estimations about the position of the landmarks. Sensor measurements are governed by a probabilistic law, often refereed as the measurement model $p(\mathbf{z}_t|\mathbf{p}_t, \mathbf{y})$, where $\mathbf{y} = \mathbf{y}_1, ..., \mathbf{y}_N$ is the set of landmarks.

Thus, most generally, SLAM is the problem of determining the localization of all the landmarks $\mathbf{y}$ and robot pose $\mathbf{p}_t$ from measurements $\mathbf{z}^t = \mathbf{z}_1, ..., \mathbf{z}_t$ and control actions $\mathbf{u}^t = \mathbf{u}_1, ..., \mathbf{u}_t$. In probabilistic terms, this is expressed by the posterior $p(\mathbf{p}_t, \mathbf{y}|\mathbf{z}^t, \mathbf{u}^t)$, where the superscript ^{t} is used to refer to a set of variables from time 1 to time t.

6.3 Filter design and implementation

As stated before, the proposed SLAM system is based on Extended Kalman filtering. The approach is a state of the art filter that takes advantage of the Homography-based odometry for both prediction stage and landmark initialization.

In Chapter 5 a loop-closing detection method based on feature matching among images of the data-base is proposed. However, this method can be used in the presented SLAM approach, because it is needed a technique for landmark data association. This procedure will not be addressed in this Chapter, and is considered to be out of the scope of the Thesis.

6.3.1 The state vector and covariance matrix

The robot pose $\mathbf{p}_t$ is comprised by the position and orientation of the vehicle at time t, so:

$$\mathbf{p}_t = [\mathbf{d}_t, \mathbf{q}_t]^T = [x, y, z, q_x, q_y, q_z, q_w]^T \tag{6.1}$$

Where $\mathbf{d}_t$ expresses the translation to the World origin encoded using cartesian representation at time t, and $\mathbf{q}_t$ is the unitary quaternion that aligns the robot to the World reference frame at time t. Using quaternions increases (in one) the number of parameters for the orientation with respect to Euler angles, but simplifies the algebra and hence, the error propagation. However, the quaternion normalization has to be taken into account after the prediction and update stages.

Landmarks will be represented by its cartesian position in 3D space:

$$\mathbf{y}_n = [x, y, z]^T \tag{6.2}$$

Thus, the state vector $\mathbf{x}_t$ is composed by the robot pose $\mathbf{p_t}$ and the set of current landmarks $\mathbf{y} = \mathbf{y}_1, ..., \mathbf{y}_n$ so:

$$\mathbf{x}_t = [\mathbf{p}_t^T, \mathbf{y}_1^T, ..., \mathbf{y}_n^T]^T \tag{6.3}$$

6.3.2 Prediction stage

Given the pose at time $t-1$, the odometer provides the translation with respect to the previous position (expressed in the $t-1$ frame) and the rotation that transforms the previous orientation into the new one (expressed in the t frame). Taking into account the quaternions algebra (see Appendix A), the state vector at time t can be computed as:

$$\mathbf{d}_t = \mathbf{d}_{t-1} + \mathbf{q}_{t-1} \otimes \dot{\mathbf{d}}_u \otimes \mathbf{q}_{t-1}^{-1} \tag{6.4}$$

$$\mathbf{q}_t = \mathbf{q}_u^{-1} \otimes \mathbf{q}_{t-1} \tag{6.5}$$

where $\mathbf{d}_u$ and $\mathbf{q}_u$ represent the estimated translation and rotation from the odometer, and $\otimes$ denotes quaternion multiplication (see Appendix A). Notice that prediction does not affect the landmark position because they are assumed to be motionless.

Computing the odometry requires to carry out the image processing between consecutive images detailed in Chapter 3: feature tracking, homography estimation and, finally, odometry. The estimated translation and rotation covariance matrices are used to compute the a priori process covariance matrix.

6.3.3 Updating stage

From the whole set of features used in the prediction stage to compute the homography, a small subset (about ten) is selected to act as landmarks. The features associated to the landmarks are taken apart and not used for the homography estimation in order to eliminate correlations among prediction and updating. For each new image, the new position of this set of features will be given by the feature tracking algorithm; this information will be used as measurement at time t, $\mathbf{z}_t$.

If the prediction stage was correct, the projection of the landmark into the camera would fit with the estimated position position of the feature given by the tracking algorithm. From the state vector, the camera is at the position $\mathbf{d}_t$ with orientation $\mathbf{q}_t$. Thus, if the landmark $\mathbf{y}_n = [x, y, z]$ corresponds with the image feature $\mathbf{m}_n = [u, v]$, and the scheme of Fig. 6.1 is considered:

$$\tilde{\mathbf{m}}_n = \mathbf{A}(\mathbf{q}_t^{-1} \otimes (\dot{\mathbf{y}}_n - \dot{\mathbf{d}}_t) \otimes \mathbf{q}_t) \tag{6.6}$$

where $\mathbf{A}$ is the camera calibration matrix and $\tilde{\mathbf{m}}_n = [\tilde{u}, \tilde{v}, h]$, so the feature position is computed as $\mathbf{m}_n = [\tilde{u}/h, \tilde{v}/h]$.

This measurement equation is applied to all the features correctly tracked from the previous image to the current one. The data association problem is solved by means of the feature matching algorithm. When the feature tracking algorithm loses a feature, it is automatically removed from the SLAM filter and the new feature, provided by the tracker, initialized. If the corresponding landmarks are well conditioned, the measurement equation constraints the current position and orientation of the UAV.

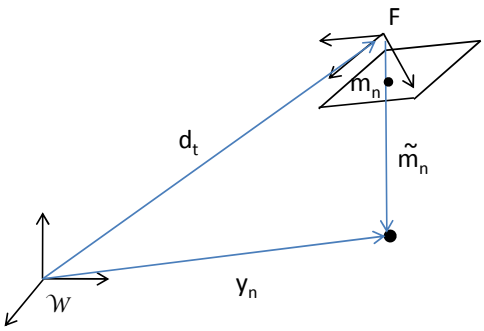


Figure 6.1: Projection of a landmark into the camera.

6.3.4 Filter and landmarks initialization

The filter state vector will be initialized to a given position and orientation. This information can be provided by external devices such as GPS and IMU, and the process covariance matrix to the corresponding error information. The position can be also initialized to zero, so the first position is assumed as origin and the corresponding covariances are zero too.

For the landmark initialization a more sophisticated method is proposed. When a new image feature is selected for being a landmark in the filter, it is necessary to compute its real position in the World frame. Due to the bearing only nature of the camera, the back-projection of the feature is given by a ray defined by the camera focal point and the landmark image. The proposed technique will take advantage of knowing the normal to the scene plane and the distance from the UAV to the ground. With this information the ground can be approximated by a plane and the landmark position as the intersection of the back-projection ray with this plane, as shown in Fig. 6.2.

Assuming that the World frame is aligned with the camera frame, the back-projection of the feature $\mathbf{m}_n = [u, v]$ will be the ray $\mathbf{r}$ defined by:

$$\mathbf{r} : \lambda \mathbf{A}^{-1} \tilde{\mathbf{m}}_n \tag{6.7}$$

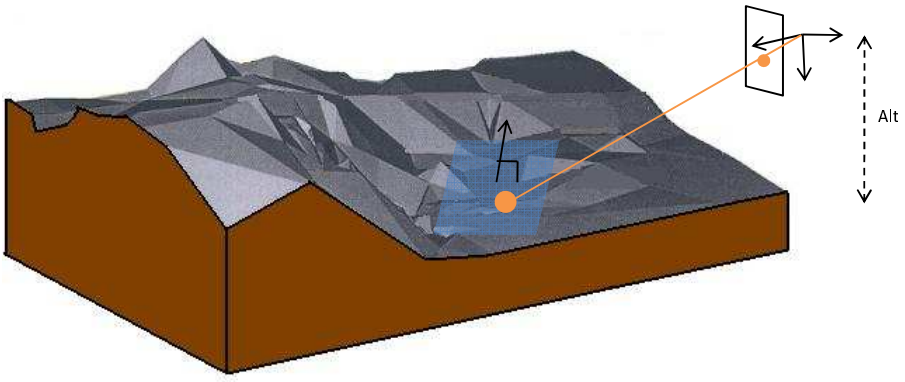


Figure 6.2: Landmark initialization representation

Where λ is an unknown variable, $\mathbf{A}$ is the camera calibration matrix and $\tilde{\mathbf{m}}_n = [u, v, 1]$. In addition, the odometer provides an estimation of the normal to the scene plane at time t denoted as $\mathbf{n}_t$. Given the distance to the plane $dist_t$, the plane Π is defined as:

$$\mathbf{\Pi} : dist_t - \mathbf{n}_t^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6.8}$$

Then, the landmark position will be computed as the intersection of the ray $\mathbf{r}$ with the plane $\mathbf{\Pi}$. If (6.7) and (6.8) are merged, the value of λ can be easily computed as:

$$\lambda = (\mathbf{n}_t^T \mathbf{A}^{-1} \tilde{\mathbf{m}}_n)^{-1} dist_t \tag{6.9}$$

and the landmark can be computed as:

$$\mathbf{y}_n = (\mathbf{n}_t^T \mathbf{A}^{-1} \tilde{\mathbf{m}}_n)^{-1} dist_t \mathbf{A}^{-1} \tilde{\mathbf{m}}_n \tag{6.10}$$

But this landmark is expressed in the camera coordinate frame. The UAV current position $\mathbf{d}_t$ and orientation $\mathbf{q}_t$ are finally used to express the landmark in the World frame:

$$\mathbf{y}_n = \mathbf{d}_t + \mathbf{q}_t \otimes ((\mathbf{n}_t^T \mathbf{A}^{-1} \tilde{\mathbf{m}}_n)^{-1} dist_t \mathbf{A}^{-1} \tilde{\mathbf{m}}_n) \otimes \mathbf{q}_t^{-1} \tag{6.11}$$

There is a strong dependence of this approach on the planarity of the scene. The more planar the scene is, the better the plane approximation, leading to smaller noise in the plane normal estimation. Thus, the landmark is very well conditioned. However, when the scene is barely approximated by a plane, the errors in the plane normal increase and the landmark estimation worsens.

6.4 Experimental results

6.4.1 Homography-based approach

To test the proposed approach, the HERO helicopter (introduced in Chapter 3) has been used. The image sequence was gathered at 15 meters of altitude with respect to the ground and with the camera pointed 45 degrees with respect to the helicopter horizontal. Mosaic-based localization cannot be carried out because the camera is not perpendicular to the ground.

The image sequence is composed by 600 samples captured at $10Hz$, so the test took 1 minute. The SLAM filter was initialized with the IMU information for the orientation. It is important to remark that no close-loop was carried out during the experiment, although there are some loops present in the UAV trajectory, because it is out of the Thesis scope. It means that the result can be improved if a reliable data association algorithm is used for detecting and associating landmarks in the filter. The complete size of the trajectory is about 90 meters long.

The results of the experiment are shown in Fig. 6.3, where the XY trajectory is plotted together with the XY estimation. More details are shown in Fig. 6.4, where the estimation in each axis and the errors are plotted. It can be seen how the uncertainty estimation is coherent with the errors. However, the position slowly diverges through time due to the absence of large loop closing. The instant orientation is not plotted because it is inherently taken into account in the computation of position.

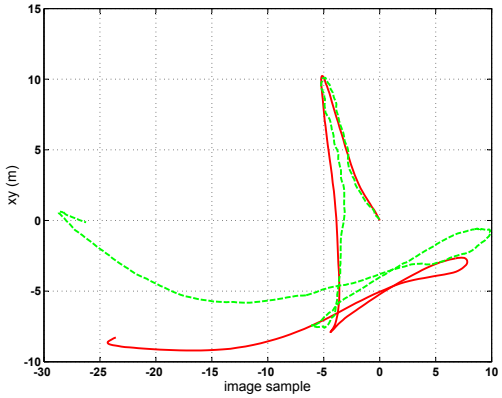


Figure 6.3: XY position estimation using the SLAM approach (green dashed line) and DGPS estimation (red solid line).

6.4.2 Inertial measurement unit (IMU) data inclusion

If the measures of the IMU are incorporated into the SLAM approach, the errors introduced by the orientation estimation can be reset, and then the localization can be partially improved. This can be easily carried out in the estimation stage of the EKF where, instead of incorporating a relative rotation, the complete orientation is provided.

This approach has been tested with the same data set. Fig. 6.5 shows the estimation compared to the DGPS measurement. It can be seen that the errors in Z and Y are significantly smaller while in X are slightly smaller with respect to the approach without considering the IMU. The XY estimation is plotted in Fig. 6.6.

6.5 Conclusions

The Chapter proposes a SLAM technique to provide localization when the mosaic-based technique cannot be performed. An Extended Kalman filter based SLAM is successfully used to compute the localization and mapping.

Two basic contributions to the SLAM with UAVs are proposed. First, the use of a vision based odometry as main motion hypothesis for the prediction stage of the Kalman filter and, second, a new landmark initialization technique that exploits the

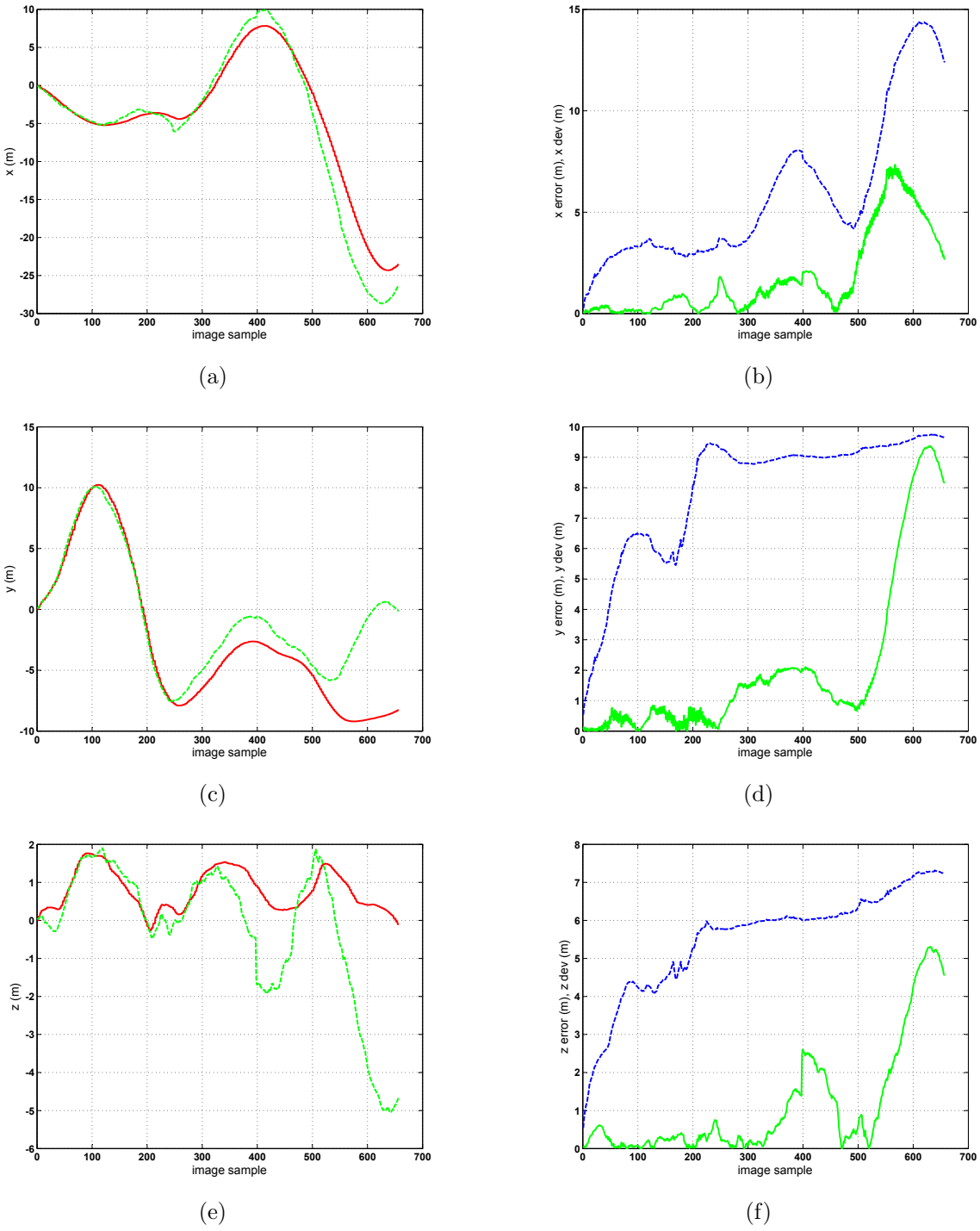


Figure 6.4: Left: Position estimation using the SLAM approach (green dashed line) and DGPS estimation (red solid line). Right: Error of the SLAM approach (green solid line) and estimated standard deviation (blue dashed line).

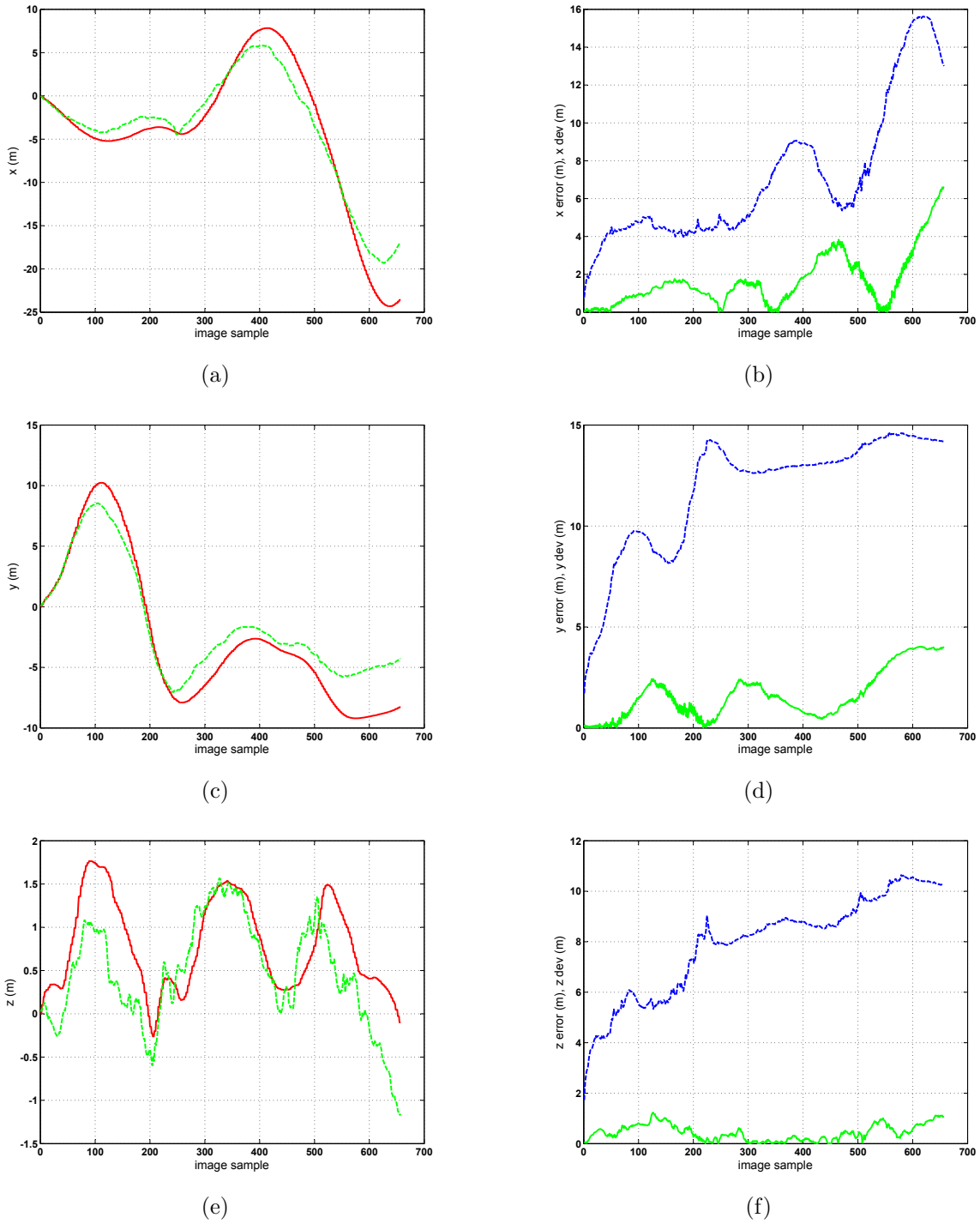


Figure 6.5: Left: Position estimation using the SLAM approach with IMU corrections (green dashed line) and DGPS estimation (red solid line). Right: Error of the SLAM approach with IMU corrections (green solid line) and estimated standard deviation (blue dashed line).

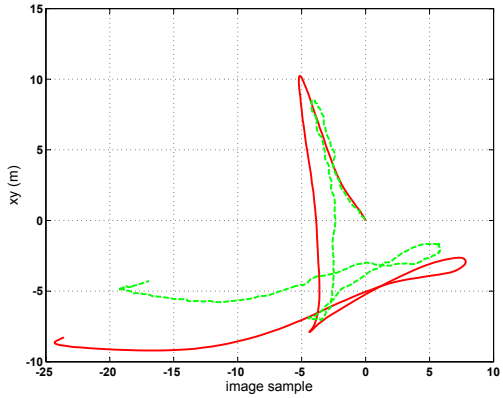


Figure 6.6: XY position estimation using the SLAM approach with IMU corrections (green dashed line) and DGPS estimation (red solid line).

benefits of estimating the normal to the scene plane. Both techniques are implemented and validated with a real UAV.

Although no large loops are closed in the experiments, the estimated position and covariance are coherent, so the result could be improved if a reliable data association algorithm is used for detecting and associating landmarks in the filter.

Chapter 7

Conclusions and future developments

This Chapter presents the Thesis conclusions and future developments. A summary of the main contributions of the Thesis and an analysis of the achieved objectives are firstly described. Then, future research activities that could extend the work presented in this document are detailed.

7.1 Summary of contributions

The aim of this Thesis is the development of algorithms for self-localization of an unmanned aerial vehicle using monocular image sequences. Monocular vision is a good solution because it offers low weight, and good accuracy and scalability. However, the scale ambiguity in monocular vision needs to be addressed. For this purpose, this Thesis uses an altitude initial estimation that can be obtained by means of different methods such as range sensor, GPS last reliable altitude measure or barometric sensor.

An important aspect of this Thesis is the use of natural landmarks instead of beacons or visual references with known positions. A general-purpose feature tracking has been used for this purpose. Although natural landmarks increase the applicability

of the proposed techniques, they also increase the complexity of the problem to be solved. In fact, outlier rejection is needed to avoid distortions.

Projective Geometry is the background theory of the proposed methods. Homographic transformations have been used to model the apparent motion between sequenced images and as a basic tool for most of the techniques proposed in this Thesis. A robust technique for homography computation has been described and used all along the experiments presented in the Thesis with good results. The study of the uncertainties associated to this estimation has also been addressed. In addition, the homography estimation method has been extended with a hierarchical method that allows the computation of the homographic model in pseudo-planar scenes.

The Thesis has proposed a visual odometry system for UAVs based on monocular imagery and a range sensor. The homography decomposition was used to extract the real camera motion and the normal vector to the scene plane. The experimental results with real UAVs showed the feasibility of the approach, at least with images taken from 15 to 150 meters, and fast, up to $10Hz$ with $320x240$ resolution images.

The Thesis proposes a localization method based on online mosaikcing has been proposed to reduce the odometry errors. The mosaic has demonstrated to be a good representation of the environment of an UAV flying at relatively high altitude.

In addition, a new technique for mosaic building that takes into account the stochastic nature of the relations among the component images has been proposed. This technique allows to extend the use of the mosaic-based localization to lower altitudes, where the parallax effect introduces significant distortion in the mosaic, and takes advantage of the loop-closing present when the UAV visits areas previously inserted in the mosaic to better estimate the relations among component images.

Mosaic-based localization is very suitable when the UAV carries out monitoring or surveillance tasks, because in many applications the UAV is expected to repeatedly visit the same areas. The stochastic framework allows to improve the mosaic estimation each time a loop is closed and, indirectly, improve the mosaic-based localization.

However, mosaic-based localization cannot be performed when the distance to the ground severely changes or the camera orientation is significantly oblique with respect to the ground. For more general situations, a SLAM technique based on Extended

Kalman filtering has been proposed. The homography-based odometry is used in this technique to make reliable motion hypothesis. A new landmark initialization technique that exploits the benefits of estimating the normal vector to the scene plane has also been proposed.

7.2 Future developments

7.2.1 Vision-based UAV control application

The first objective of the author in the near future is to apply the odometry and localization techniques proposed in this Thesis to close the position control loop in UAVs. These algorithms will be used to estimate the motion of the HERO helicopter in several situations such as hovering, landing or take-off.

These applications will provide positioning information without the use of special positioning devices, which is very important in disaster scenarios where a fast system deploy is required and the GPS may not be suitable.

7.2.2 Simultaneous tracking and homography computation

Another research goal will be to reduce the computation load needed for the real camera motion estimation. Although the algorithm presented in Chapter 3 works well, feature tracking and homography computation algorithms are needed to later compute the motion from the homographies. If a simultaneous computation of the homography and the feature tracking were possible, the computation load could be significantly reduced.

The homography models feature motion from the previous image to the current one. If a well distributed set of matches, four at least, were provided by a feature tracking algorithm, a homography could be computed. Although the estimated homography is inaccurate when it is computed with only a few matches, it probably describes most of the transformation among the images. Thus, this homography can be used to determine the searching areas for a subset of new features. Once these new features are matched, they can be used to re-estimate the homography, this time with

more information. The growth in the number of matches will increase the accuracy of the estimated homography and then, reduce the searching area for new features. This process can be repeated until the tracking of all the features has been performed.

Moreover, if the identity (3.9) is considered, the homography can be replaced with an expression that is function of the translation, rotation and plane normal. As a result, a simultaneous feature tracking and real camera motion could be carried out in a easy iterative method.

7.2.3 Wall inspection

An interesting application of the homography-based odometry is the relative position estimation with respect to a wall. The GPS estimation is usually inaccurate when the UAV flies close to buildings. The homography-based odometry could be valuable tool to perform approach maneuvers, or for data interpretation and processing.

The homography-based odometer is able to obtain the relative orientation of the scene plane with respect to the camera. This information can be used to control the helicopter in order to obtain a perpendicular view of the walls of the inspected building. Given an initial estimation of the distance to the wall, it is also possible to obtain UAV position with respect to the plane each time a new image is received.

Moreover, the knowledge of normal vector to the plane can be used to generate virtual orthogonal views of the inspected building. The orthogonal view can be an important tool when it is necessary to recognize geometric features, because variations in the camera orientation may create significant distortion.

7.2.4 Navigation based on geo-referenced mosaics

The application of mosaic-based localization described in Chapter 4 with geo-referenced mosaic will be researched. The idea is to use pre-computed mosaics with localization references in order to estimate the position of the UAV.

The images gathered by the camera onboard the vehicle are matched with the mosaic to compute their exact location. Then, if the mosaic is geo-referenced, the relation with the mosaic can be used to estimate the real position of the UAV. This

application is really interesting when the UAV performs monitoring or survey tasks in which the vehicle will move over the same area all the time.

Appendix A

Quaternions

A.1 Definition

Quaternions are a non-commutative extension of complex numbers. They were first described by the Irish mathematician Sir William Rowan Hamilton in 1843 and applied to mechanics in three-dimensional space. At first, quaternions were regarded as pathological, because they disobeyed the commutative law. Although they have been superseded in most applications by vectors, they still find uses in both theoretical and applied mathematics, in particular for calculations involving three-dimensional rotations, such as in 3D computer graphics.

As Hamilton pointed out in 1833, the addition symbol used in the Cartesian representation of a complex number $a + ib$ is somewhat misleading, since a real and purely imaginary number cannot be directly added together arithmetically. A more suitable representation might be as an ordered pair of real numbers $[a, b]^T$ together with a set of manipulation rules that define how to perform operations like addition and multiplication of these pairs.

It seems natural, then, to speculate whether there might be some form of extended number system whose numbers may be interpreted as points in three-dimensional space, with a corresponding representation as number triples.

The simplest such extension would seem to be numbers of the form $a + ib + jc \equiv [a, b, c]^T$ where i and j are distinct, independent square roots of -1. Hamilton attempted to define operations on these triples that were analogous to those on complex numbers. Addition and subtraction are naturally implemented as component-wise operations on the three real numbers. Multiplication, however, presented a problem.

In three dimensions it is needed two parameters to specify the direction of the axis for a rotation, a third to specify the angle of rotation, and yet another to determine a scaling for the length. Thus, it would be needed to specify four parameters and will be represented by a four dimensional vector such as:

$$w + xi + yj + zk \equiv [x, y, z, w]^T \equiv [\mathbf{v}, w]^T \tag{A.1}$$

Where $\mathbf{v}$ is a vector expressed in the ijk basis.

A.2 Quaternion algebra

Given the following quaternions:

$$\mathbf{a} \equiv a_w + a_x i + a_y j + a_z k \equiv [\mathbf{a}_v, a_w]^T \equiv [a_x, a_y, a_z, a_w]^T \tag{A.2}$$

$$\mathbf{b} \equiv b_w + b_x i + b_y j + b_z k \equiv [\mathbf{b}_v, b_w]^T \equiv [b_x, b_y, b_z, b_w]^T \tag{A.3}$$

A.2.1 Basis multiplication

The set of equations:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{A.4}$$

is the fundamental formula for quaternion multiplicative identities, summarized in the following multiplication table of basis quaternions:

$$\begin{array}{rclcrcl} ij & = & k, & & ji & = & -k \\ jk & = & i, & & kj & = & -i \\ ki & = & j, & & ik & = & -j \end{array} \tag{A.5}$$

A.2.2 Addition

Addition is the simple map of the addition operator over each element in the quaternions:

$$\mathbf{a} + \mathbf{b} \equiv [a_x + b_x, a_y + b_y, a_z + b_z, a_w + b_w]^T \tag{A.6}$$

A.2.3 Substraction

Addition is the simple map of the substraction operator over each element in the quaternions:

$$\mathbf{a} - \mathbf{b} \equiv [a_x - b_x, a_y - b_y, a_z - b_z, a_w - b_w]^T \tag{A.7}$$

A.2.4 Multiplication

The multiplication is the exception in the quaternion algebra and is normally denoted by the symbol $\otimes$. It is computed as follows:

$$\mathbf{c} = \mathbf{a} \otimes \mathbf{b} = a_w b_w - \mathbf{a}_v \mathbf{b}_v + a_w \mathbf{b}_v + b_w \mathbf{a}_v + \mathbf{a}_v \times \mathbf{b}_v \tag{A.8}$$

Or, in an optimized way:

$$\begin{aligned} c_x &= a_w b_x + a_x b_w + a_y b_z - a_z b_y && \text{(A.9)} \\ c_y &= a_w b_y - a_x b_z + a_y b_w + a_z b_x && \text{(A.10)} \\ c_z &= a_w b_z + a_x b_y - a_y b_x + a_z b_w && \text{(A.11)} \\ c_w &= a_w b_w - a_x b_x - a_y b_y - a_z b_z && \text{(A.12)} \end{aligned}$$

Notice that the multiplication is not commutative, so in general $\mathbf{a} \otimes \mathbf{b} \neq \mathbf{b} \otimes \mathbf{a}$

A.2.5 Inverse (Conjugation)

The quaternion conjugate corresponds to the negation of each of the elements that would have a spatial representation, which are the elements in the i basis, the j basis and the k basis.

The usual symbol for the conjugation of the quaternion $\mathbf{a}$ is $\mathbf{a}^*$. Thus, the conjugate can be written as:

$$\mathbf{a}^* \equiv a_w - a_x i - a_y j - a_z k \equiv [-\mathbf{a}_v, w]^T \equiv [-a_x, -a_y, -a_z, a_w]^T \tag{A.13}$$

If the quaternion represents a rotation, its conjugate expresses the inverse rotation. It is the reason that the conjugate is usually refereed as inverse and denoted as $\mathbf{a}^{-1}$

A.3 Quaternion conversion

This section details the steps to convert from quaternion to rotation matrix representation and vice versa.

A.3.1 Quaternion to rotation matrix

The rotation matrix $\mathbf{R}$ corresponding to the quaternion $\mathbf{q} = [q_x, q_y, q_z, q_w]^T$ can be computed as follows:

$$\mathbf{R} = (q_w^2 - \check{\mathbf{q}}^T\check{\mathbf{q}})\mathbf{I}_3 + 2\check{\mathbf{q}}\check{\mathbf{q}}^T - 2q_w\mathcal{Q} \tag{A.14}$$

With $\mathbf{I}_3$ the 3×3 identity matrix, and

$$\check{\mathbf{q}} = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}, \quad \mathcal{Q} = \begin{bmatrix} 0 & -q_z & q_y \\ q_z & 0 & -q_x \\ -q_y & q_x & 0 \end{bmatrix} \tag{A.15}$$

A.3.2 Rotation matrix to quaternion

Given the rotation matrix $\mathbf{R}$:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{A.16}$$

Algorithm 2 Rotation matrix to quaternion conversion

$T = 1 + r_{11} + r_{22} + r_{33}$
if $T > 0.00000001$ **then**
  $k = 2\sqrt{T}$
  $q_x = (r_{23} - r_{32})/k$
  $q_y = (r_{31} - r_{13})/k$
  $q_z = (r_{12} - r_{21})/k$
  $q_w = 0.25k$
else if $r_{11} > r_{22}$ AND $r_{11} > r_{33}$ **then**
  $k = 2\sqrt{1.0 + r_{11} - r_{22} - r_{33}}$
  $q_x = 0.25k$
  $q_y = (r_{12} + r_{21})/k$
  $q_z = (r_{31} + r_{13})/k$
  $q_w = (r_{23} - r_{32})/k$
else if $r_{22} > r_{33}$ **then**
  $k = 2\sqrt{1.0 + r_{22} - r_{11} - r_{33}}$
  $q_x = (r_{12} + r_{21})/k$
  $q_y = 0.25k$
  $q_z = (r_{23} + r_{32})/k$
  $q_w = (r_{31} - r_{13})/k$
else
  $k = 2\sqrt{1.0 + r_{33} - r_{11} - r_{22}}$
  $q_x = (r_{31} + r_{13})/k$
  $q_y = (r_{23} + r_{32})/k$
  $q_z = 0.25k$
  $q_w = (r_{12} - r_{21})/k$
end if

Algorithm 2 shows the pseudocode to optimally compute the corresponding quaternion $\mathbf{q} = [q_x, q_y, q_z, q_w]^T$.

A.4 Rotation of a vector

Quaternions can be used to rotate a vector. Thus, given a 3D vector $\mathbf{v}$ and a rotation expressed as a quaternion $\mathbf{q}$, the rotated vector $\mathbf{r}$ is computed as follows:

$$\mathbf{r} = \mathbf{q} \otimes \dot{\mathbf{v}} \otimes \mathbf{q}^{-1} \tag{A.17}$$

Where $\dot{\mathbf{v}} = [\mathbf{v}, 0]^T$ and the quaternion has norm one. This can easily be realized and is most faster than the transformation using a rotation matrix.

It is important to remember that successive rotations $\mathbf{R}_1$ and $\mathbf{R}_2$ are combined as $\mathbf{R}_3 = \mathbf{R}_2\mathbf{R}_1$ but quaternions have inverse ordering of multiplication when compared to rotation matrix multiplication and hence $\mathbf{q}_3 = \mathbf{q}_1 \otimes \mathbf{q}_2$.

Appendix B

Breve resumen de la Tesis

B.1 Motivaciones

Las aplicaciones de robots en exteriores requieren de una mayor movilidad que la provista por la mayor parte de los vehículos terrestres. De hecho, a pesar de los numerosos adelantos tecnológicos producidos en los últimos 20 años en los vehículos autónomos terrestres, la navegación en entornos desconocidos aún supone grandes retos.

Los vehículos autónomos aéreos o UAVs (acrónimo inglés) son una buena opción como sistema robótico debido a que no están afectados por estas limitaciones. Además, los últimos adelantos tecnológicos han permitido incrementar las capacidades de autonomía de estos vehículos, tanto en procesamiento a bordo, como en tiempo de vuelo. Así, hoy en día existen UAVs en los que se llevan a cabo tareas automáticas de percepción del entorno, estimación de la posición, navegación reactiva o desarrollo de tareas complejas en entornos desconocidos.

En la mayoría de los casos, los UAVs utilizan el Sistema de Posicionamiento Global (GPS en inglés) para la estimación de la posición del vehículo. Sin embargo, como apunta el informe Volpe (Volpe, 2001), dicha estimación no está exenta de errores. Los sistemas GPS dependen en gran medida del número de satélites utilizados para el cálculo de la posición y de la calidad de la señal recibida en el dispositivo. Así, efectos

radioeléctricos como el multi-camino o la polarización de la antena pueden generar una reducción importante en la calidad de la estimación.

Estos problemas ocasionales en la estimación de la posición han sido tradicionalmente resueltos mediante el uso de la odometría del robot. Ésta consiste en calcular la posición relativa del robot con respecto a un punto de partida mediante sensores locales colocados en las ruedas, normalmente velocidad y orientación. Si un sistema de odometría está presente en el robot, éste puede ser utilizado como sistema de posicionamiento de respaldo cuando la precisión en las estimaciones del GPS se reduce hasta niveles críticos. Por desgracia, los vehículos aéreos suelen carecer de odometría debido fundamentalmente a la falta de un sensor que permita estimar el desplazamiento relativo del vehículo con respecto al suelo. Así, el estudio en técnicas de odometría para UAV resulta de especial interés.

En el caso concreto de pequeños UAVs, su baja capacidad de carga impone serias restricciones en cuanto a los tipos de sensores que pueden ser utilizados para la odometría. Los sensores basados en laser 2D o 3D son muy pesados y tienen una importante dependencia con la distancia hasta el suelo. Los dispositivos que permiten calcular la profundidad, aunque existen en tamaño pequeño, suelen tener un rango de actuación muy corto, normalmente entre los 15 o 20 metros. La visión estéreo, que se ha utilizado con buenos resultados en vehículos aéreos dado su bajo peso y versatilidad, tiene como inconveniente que la distancia entre cámaras impone duras restricciones en el rango de alturas de vuelo.

Descartadas las anteriores opciones, los sistemas de visión monoculares parecen ser la mejor solución en términos de peso, precisión y escalabilidad. Sin embargo, a pesar de haber sido presentados diversos métodos basados en balizas visuales con posiciones conocidas, el estudio basado en marcas naturales sigue siendo un gran reto apenas tratado por la comunidad científica.

Esta Tesis propone un sistema de visión monocular para el cálculo de la odometría y sistemas de localización basados en visión, que actúen como respaldo frente a posibles errores del GPS. Así, se utilizarán técnicas de visión artificial basada en las

imágenes tomadas por una cámara montada en el UAV para el cálculo de la rotación y la traslación relativa, y para la localización del vehículo. Este análisis tendrá en cuenta el carácter estocástico de las estimaciones y diversas consideraciones prácticas.

B.2 Antecedentes

Las primeras investigaciones en visión aplicada a la estimación de la posición de UAVs fueron llevadas a cabo en la Universidad Carnegie-Mellon (CMU). En (Amidi et al., 1998), Amidi describe un sistema de odometría basado en visión que permitía seguir objetivos terrestres y percibir el desplazamiento y velocidad relativas del UAV mediante visión estereoscópica. Las mismas técnicas de seguimiento visual junto con sensores inerciales, fueron utilizadas para el seguimiento de una trayectoria, aterrizaje y despegue. El helicóptero autónomo del CMU demostró también sus capacidades para el seguimiento de objetivos móviles mediante el uso de una electrónica especialmente diseñada alojada en el vehículo.

La visión estereoscópica se utiliza también en el helicóptero GTMax de la universidad Georgia Tech (Johnson and Schrage, 2003) donde el sistema VISTA es usado para le detección y evitación de obstáculos en tiempo real utilizando un pequeño sistema estereoscópico comercial. El método, descrito en (Byrne et al., 2006), combina visión estéreo con una fase de segmentación global para incrementar la robustez ante correspondencias erróneas, dando lugar a un práctico sistema de evitación de obstáculos.

Los sistemas basados en visión también han sido muy utilizados para el aterrizaje de vehículos autónomos. El proyecto BEAR de la Universidad de Berkeley es un buen ejemplo, así, en (Shakernia et al., 2002; Vidal et al., 2002) se detalla la estimación de posición basado en visión para vehículos aéreos relativos a un objetivo plano y el aterrizaje. Se usa un método basado en la geometría de múltiples vistas de la misma escena plana para calcular el movimiento del UAV con respecto el objeto plano.

En (Garcia-Pardo et al., 2001) se describe un algoritmo para el aterrizaje basado en la búsqueda de áreas planas mediante procesamiento de imágenes. Métodos basados en visión para el aterrizaje en pista con características conocidas se describen en

(Saripalli et al., 2003; Saripalli and Sukhatme, 2003), donde también se considera el aterrizaje en una plataforma que se mueve lentamente.

Los métodos que consideran la localización y mapeo simultáneo (SLAM) están muy relacionados con el problema de la localización. Aunque el SLAM basado en visión ha sido profundamente estudiado para el caso de vehículos terrestres y ha demostrado una gran utilidad para la percepción del entorno y estimación consistente de la posición del robot, pocas han sido las aplicaciones en vehículos aéreos. Cabe mencionar las investigaciones realizadas por el laboratorio LAAS en Francia y por el Centre for Autonomous System en Australia. Así, LAAS ha desarrollado un sistema estéreo para el dirigible KARMA (Lacroix et al., 2002; Hygounenc et al., 2004) donde utilizan algoritmos para el seguimiento de correspondencias entre imágenes y filtros de Kalman para la localización y mapeo simultaneo, con muy buenos resultados. Sin embargo, este método no es adecuado para helicópteros debido a que la distancia máxima entre cámaras es reducida y por tanto solo podría ser utilizada a baja altura. También se utiliza SLAM con visión en la plataforma de ala fija Delta en los trabajos desarrollados por Kim y Sukkarieh (Kim and Sukkarieh, 2004) en el Centre for Autonomous Systems. Se distribuye un conjunto de puntos de referencia artificiales y de tamaño fijo en el suelo con la idea de ser fácilmente detectados y localizados por el hardware a bordo del vehículo, dando lugar a un sensor de distancia/orientación/elevación.

El uso de cámaras omnidireccionales para el control del vehículos aéreos se ha considerado en (Hrabar and Sukhatme, 2003). La cámara se utiliza para hacer permanecer el helicóptero en el centroide de un conjunto de referencias visuales artificiales. El artículo muestra la viabilidad del método, pero no se detallan pruebas con el algoritmo de control. Cámaras omnidireccionales han sido utilizadas también en (Demonceaux et al., 2006) para estimar orientación del UAV. El método se basa en separar mediante procesamiento digital de imágenes la línea del horizonte. A pesar de parecer una técnica robusta, los autores no comparan los resultados con ninguna referencia de orientación y por tanto no es posible evaluar la precisión de la estimación.

Algoritmos de visión artificial son utilizados para el seguimiento de rasgos de edificios en el trabajo presentado en (Mejías et al., 2006). Estos rasgos junto con las

medidas de GPS son utilizados para mantener el UAV alineado con un conjunto de rasgos seleccionados.

Finalmente, en (Garcia et al., 2002) se han propuesto algoritmos para la localización de vehículos autónomos submarinos o AUVs (acrónimo Inglés) basada en visión monocular y construcción de mosaicos. A pesar de ser un entorno completamente diferente, las limitaciones son parecidas a las presentes en los UAVs. El método utiliza mosaicos construidos en línea para crear un modelo plano del fondo submarino y utilizar así las relaciones entre imágenes para calcular la traslación del vehículo. El concepto es original y de gran aplicabilidad, pero el sistema solo permite calcular la traslación del vehículo.

B.3 Objetivos de la Tesis

El principal objetivo de esta Tesis es el desarrollo de técnicas de visión artificial para localizar un UAV por medio de las imágenes tomadas por una cámara montada en el mismo. Es importante señalar que se utilizarán las características naturales del entorno para este propósito, en lugar de balizas visuales con posiciones conocidas. Los algoritmos tanto de odometría como de localización del UAV están fundamentados en la teoría de Geometría Proyectiva. Ésta ha sido utilizada con mucho éxito para expresar el proceso de proyección de imágenes en la cámara y en numerosos problemas relacionados con la visión por computador tales como la visión estereoscópica (Faugeras and Luong, 2001), proyección inversa de imágenes, geometría de múltiples vistas de la misma escena (Hartley and Zisserman, 2004) o calibración de cámaras (Zhang, 1999).

En este sentido, la Tesis primero lleva a cabo una introducción a la Geometría Proyectiva y, más concretamente, al cálculo de homografías en el Capítulo 2. La Homografía se utilizará como representación formal del movimiento entre dos imágenes consecutivas cuando la escena visualizada puede ser aproximada por un plano. Los resultados experimentales presentados en este documento demostrarán que dicha aproximación es válida cuando el UAV vuela relativamente alto y se propondrán mecanismos para extender dicha aproximación a altitudes medias. El algoritmo para el

cálculo de homografías se ha probado con miles de imágenes tomadas por diferentes UAVs y a diferentes alturas, desde quince a ciento cincuenta metros. Este algoritmo se describió brevemente en (Ollero et al., 2004) y es parecido al método propuesto por Bosch en (Bosch et al., 2006). Esta técnica se ha probado con buenos resultados durante los experimentos generales del proyecto COMETS (Ollero et al., 2005) para la compensación del movimiento en las imágenes tomadas por los UAVs (Merino et al., 2006a). El capítulo también describe el cálculo de la matriz de covarianza asociada a la homografía estimada. Esta información será crítica a lo largo de la Tesis para poder propagar y estimar los errores en aquellos procesos en los se utilice la homografía.

El Capítulo 3 detalla un método que permite estimar el movimiento real que se produce en una cámara por medio del análisis de la homografía que relaciona imágenes consecutivas. El Capítulo extiende el trabajo presentado en (Caballero et al., 2005), donde se presenta una de las primeras aplicaciones de la homografías para la estimación del movimiento de un UAV. A pesar de que el trasfondo teórico del método fue desarrollado durante la década de los ochenta por Tsai (Tsai et al., 1982), la particularización del método al entorno de UAVs y el estudio de las incertidumbres de las estimaciones suponen grandes avances. Imágenes y telemetría de UAVs reales se utilizan en este Capítulo para poder validar el algoritmo. Los resultados experimentales muestran que, a pesar de generar buenas estimaciones, se produce un efecto acumulativo del error en la estimación de la posición del UAV, que acaba por hacer diverger la estimación.

El Capítulo 4 presenta una nueva técnica para reducir el impacto de la acumulación del error en la estimación de la posición. El Capítulo describe un método para construir mosaicos en línea a partir de las homografías que relacionan imágenes consecutivas. El mosaico se utiliza para poder construir un modelo del entorno consistente y, de este modo, ser capaz de detectar derivas en la estimación de la posición. Esta técnica, presentada por primera vez en (Caballero et al., 2006) y junto con la estimación de posición del Capítulo 3 en (Merino et al., 2006b), es especialmente adecuada para llevar a cabo tareas de monitorización y vigilancia, donde el UAV pasará repetidamente por encima de las mismas áreas. El Capítulo muestra resultados

experimentales donde se hacen evidentes los beneficios de la localización basada en mosaico frente a la aproximación odométrica.

Sin embargo, la naturaleza estática del mosaico construido utilizando el método descrito en el Capítulo 4 hace imposible la actualización del mosaico cuando el UAV visita zonas ya registradas. El Capitulo 5 trata con este problema, propone un nuevo marco estocástico donde la matriz de covarianza que relaciona dos imágenes consecutivas se utiliza para poder modelar y propagar los errores relacionados con el posicionamiento de la imágenes. De esta manera, cuando el UAV visite una zona anteriormente introducida en el mosaico, la información del cierre del bucle puede ser utilizada para refinar el posicionamiento entre imágenes. Además, el Capítulo 5 propone un nuevo método para el cálculo de la homografía, basada en una jerarquía de modelos proyectivos, que permite extender su cálculo a secuencias de imágenes tomadas a media/baja altura. El Capítulo 5 detalla el trabajo presentado en (Caballero et al., 2007) y aporta nuevos resultados experimentales.

Los Capítulos 4 y 5 consideran que el vuelo llevado a cabo por el UAV es tal que permite construir un mosaico a la vez que se mueve, este mosaico es utilizado para modelar el entorno para así reducir los errores en la localización. Las mayores restricciones de este algoritmo son las relacionadas con la posición de la cámara y la altura del UAV: la cámara debe estar lo más perpendicular posible a la escena y la altura debe permanecer aproximadamente constante para garantizar la coherencia del mosaico. El Capítulo 6 considera este problema y lo resuelve mediante la aplicación de técnicas clásicas de SLAM. El Capítulo también considera el uso de sensores de medida inercial para reducir los errores de orientación y así mejorar la estimación de la localización.

Finalmente, el Capítulo 7 resume las contribuciones de la Tesis y describe los siguiente pasos a llevar a cabo por el autor en cuanto a localización basada en secuencias de imágenes monoculares se refiere.

B.4 Resumen del Capítulo 2

Este capítulo describe brevemente la herramienta matemática que se utiliza a lo largo de toda la Tesis, la Geometría Proyectiva (Hartley and Zisserman, 2004; Faugeras and Luong, 2001). En particular, se introduce la notación a utilizar y las transformaciones proyectivas del plano. Estas transformaciones permitirán modelar las distorsiones geométricas producidas en un plano cuando se toma una imagen de éste con una cámara de perspectiva.

Además se detalla el proceso de estimación de la matriz de homografía, la cual permite relacionar dos imágenes tomadas de una misma escena plana. Dicha estimación se basa en la información aportada por la posición de un conjunto de correspondencias entre ambas imágenes.

En la estimación de la homografía es necesario, en primer lugar, filtrar los datos de las posiciones de las correspondencias de modo que se puedan eliminar errores de seguimiento o correspondencias asociadas a elementos no planos. Posteriormente, se aplican técnicas numéricas para la obtención de la matriz que mejor se ajuste a los datos.

Para poder filtrar los datos de entrada se aplica el algoritmo de la mínima mediana de los mínimos cuadrados (LMedS) (Rousseeuw and Leroy, 1987). Este algoritmo permite obtener la desviación típica de los datos, con la que se establece una cota máxima de error que permite la detección de datos incorrectos.

El cálculo definitivo de la matriz de homografía se hace mediante un algoritmo de M-Estimaciones (Huber, 1981), que sirve para establecer con gran precisión el valor de la matriz de homografía a partir de los valores ya filtrados. Este algoritmo también permite detectar errores en el espacio de datos de las correspondencias.

B.5 Resumen del Capítulo 3

Los vehículos aéreos no tripulados (UAV) suelen usar solo el GPS para obtener una medida fiable de la posición en la que se encuentran. Esto puede ser un importante problema ya que la precisión del GPS depende directamente del número de satélites

utilizados para la estimación de la medida. Si la visibilidad es baja la estimación de la posición puede contener importantes errores.

Con intención de tener un sistema de posicionamiento robusto, este Capítulo presenta un conjunto de herramientas que permitan dotar al UAV de un sistema de posicionamiento mínimo cuando el error de GPS se ve incrementado por encima de ciertas cotas. Se implementa una odometría basada en sistemas de medida locales, fundamentalmente en visión monocular.

En este capítulo se describe un método basado en visión que permitirá la estimación del movimiento real de una cámara utilizando diferentes vistas de una escena plana. En capítulos anteriores se ha mostrado cómo es posible obtener el movimiento de la cámara en el sistema de referencia píxel cuando la escena es casi plana mediante el uso de homografías, sin necesidad de realizar una reconstrucción 3D.

En (Faugeras and Luong, 2001) se establece que la homografía que relaciona dos imágenes tomadas de la misma escena desde diferentes puntos de vista puede ser expresada como sigue:

$$\mathbf{H_{12}} = \mathbf{A}\cdot\mathbf{R}_2\cdot(\mathbf{I} - w\mathbf{t}_2\cdot\mathbf{n}^t)\cdot\mathbf{A}^{-1} \tag{B.1}$$

Donde $\mathbf{A}$ es una matriz $3x3$ que contiene los parámetros de calibración intrínsecos de la cámara, $\mathbf{t}_2$ es la posición de la segunda vista de la escena tomada en el sistema de referencia de la primera, $\mathbf{n}$ es un vector unitario normal al plano de la escena en el sistema de referencia de la primera vista (con sentido hacia fuera de la cámara), w es la inversa de la distancia de la primera vista hasta el plano de la escena y $\mathbf{R}_2$ es la rotación que transforma el sistema de coordenadas de la primera vista en el de la segunda.

En (Tsai et al., 1982) se muestra que, si la cámara está calibrada (la matriz $\mathbf{A}$ es conocida), entonces es posible obtener dos soluciones de $\mathbf{R}_2$, $\mathbf{t}_2$, y $\mathbf{n}_2$ a partir de $\mathbf{H}_{12}$, salvo un factor de escala. Si un tercer punto de vista es considerado, junto con su homografía $\mathbf{H}_{13}$ respecto a la primera vista, es posible obtener una solución única gracias a que la normal al plano debe ser única ((Shakernia et al., 2002)), es decir, en teoría $\mathbf{n} = \mathbf{n_2} = \mathbf{n_3}... = \mathbf{n_i}$.

Si se utiliza un sistema de medida independiente, tal como un sensor de distancias, se podría conocer el factor de escala sin más que calcular la distancia entre el primer punto de vista y el plano $(1/w)$. Este sensor puede ser un simple laser o un sensor de ultrasonidos, en general sensores de bajo coste y poco peso.

El Capítulo describe un método para el cálculo de la matriz de covarianzas de la rotación, traslación y normal al plano estimadas. Dicha matriz de covarianzas se utiliza como índice para medir la bonanza de la estimación.

A pesar de que la estimación de la traslación y rotación utilizando este método es bastante robusta y precisa, la acumulación de un alto número de estimaciones para localizar el vehículo con respecto a la posición inicial genera la acumulación de error en la localización y, por tanto, la divergencia de la posición a lo largo del tiempo.

B.6 Resumen del Capítulo 4

Un mosaico es una imagen que reproduce una escena continua de gran tamaño formada a partir varias imágenes, de menor resolución, de la misma. Las imágenes usadas para construir el mosaico pueden estar sometidas a rotaciones y traslaciones importantes respecto al marco de referencia, la primera imagen.

Una posible aplicación de la percepción en los vehículos aéreos es la generación de mosaicos de alta resolución. Estos mosaicos pueden ser creados con múltiples propósitos tales como la generación de imágenes de alta resolución, marcos de referencia para vehículos terrestres o incluso para mejorar los sistemas de posicionamiento del vehículo que lo construye u otros.

Resulta de especial interés el poder generar estos mosaicos sin más información que la dada por las imágenes tomadas por la cámara del vehículo, ya que, en este caso, toda la información aportada por el mosaico puede ser utilizada como un sensor de posicionamiento con determinadas restricciones.

El método que se presenta en este capítulo hará uso del sistema de establecimiento y seguimiento de correspondencias detallado en (Ollero et al., 2004) y de la matriz de homografía para el cálculo del movimiento entre dos imágenes consecutivas presentado en el Capítulo 3. Como principal característica se puede señalar que el método

propuesto genera el mosaico a partir de la primera pasada, por tanto, si se ve más de una vez una determinada escena, en el mosaico solo aparecerán las imágenes correspondientes a la primera vez que se visualizó. La razón de esta forma de proceder es que se simplifica el proceso de generación del mosaico, además de aumentar la exactitud del posicionamiento de las imágenes en el mismo.

El método propuesto está estructurado en torno a dos pasos de seguimiento de correspondencias por cada una de las imágenes tomadas por el sistema de percepción del vehículo. El primero de estos pasos se denomina fase de estimación del movimiento relativo y consiste en calcular el movimiento producido entre dos imágenes consecutivas. El fin de esta fase es descomponer el movimiento de una imagen en suma de pequeños movimientos independientes unos de otros.

El segundo paso de seguimiento de correspondencias está destinado a corregir el error global de posicionamiento de las imágenes dentro del mosaico. Mediante el primer paso se calcula el movimiento producido en una imagen sin tener en cuenta los errores de estimación de movimiento previos ni el mosaico en sí mismo. En esta fase se pretende detectar las diferencias de posición entre la calculada usando el movimiento relativo y la posición dentro del mosaico construido hasta el momento. Conocida la diferencia, se aplicará un factor de corrección al posicionamiento de la imagen en el mosaico.

Durante ambas fases serán necesarias funciones que permitan aplicar las transformaciones calculadas sobre las imágenes. Una primera función permitirá aplicar transformaciones sobre imágenes aplicando la técnica del pixel más parecido que se verá en secciones sucesivas. Una segunda función permitirá transformar las imágenes para introducirlas dentro del mosaico, esta función se encargará de ecualizar la imagen introducida dentro del mosaico para que no se produzcan cambios severos de iluminación a lo largo del mismo. Al mismo tiempo, esta última función se encargará de introducir datos en el mosaico solo en las partes no visitadas del mismo, ya que el mosaico es de una sola pasada.

La aplicación de esta técnica sobre el conjunto de imágenes capturadas por el UAV permite una mejora en la localización, debido, fundamentalmente, al aumento de la coherencia de las homografías entre imágenes. El Capítulo muestra resultados

experimentales en localización que mejoran considerablemente la estimación respecto a la odometría.

B.7 Resumen del Capítulo 5

El método presentado en el Capítulo 4 es robusto y permite limitar los errores en la localización del vehículo. Sin embargo, una vez que el mosaico ha sido construido con las imágenes capturadas por el UAV, no es posible actualizarlo dado que éste no posee ninguna información relacionada con la precisión de la estimación.

El presente Capítulo describe un método que permite crear mosaicos en línea de manera consistente, teniendo en cuenta el carácter estocástico de las relaciones entre imágenes. De esta manera, el mosaico deja de ser una imagen estática y se convierte en una base de datos de imágenes relacionadas unas con otras.

Este nuevo método para crear mosaicos, permite actualizar la posición de las imágenes componentes cuando nueva información sobre el posicionamiento de éstas es obtenida. Esto ocurre cuando el UAV visita una zona que ya ha registrado y cierra un bucle de posición. El bucle aporta una homografía que relaciona la imagen actualmente capturada con una de la imágenes presente en la base de datos. Dicha homografía encapsula los posibles errores de posicionamiento entre imágenes ocurridos a lo largo del bucle.

La homografía obtenida del cierre de un bucle es utilizada en un proceso de minimización, basado en filtrado de Kalman, que permite propagar los errores acumulados en esa homografía hacia atrás en las imágenes de bucle. Como consecuencia se produce un reajuste de las imágenes del mosaico acorde a la información aportada. El capítulo describe los pasos necesarios para llevar a cabo la minimización basada en filtrado de Kalman y presenta resultados de localización.

Un aspecto tratado en este Capítulo es el cálculo de la matriz de homografía cuando el vehículo vuela a media/baja altura. Esto resulta fundamental debido a que la presencia de objetos tridimensionales en la escena aumenta conforme la altura al suelo disminuye. Para solucionar este problema, el Capítulo propone una jerarquía de

modelos proyectivos que se va relajando en función de variables como el número de correspondencias entre imágenes o el número de correspondencias incorrectas.

B.8 Resumen del Capítulo 6

Este Capítulo extiende el proceso de localización utilizando secuencias de imágenes monoculares a una situación más general, en la cual es posible que no se pueda llevar a cabo la construcción del mosaico. Las razones que dificultan dicha construcción pueden ser varias, en general, si la cámara está orientada de forma no perpendicular al suelo o el UAV cambia de altura con gran frecuencia, el mosaico generado puede llegar a ser incoherente.

Para generalizar el problema de la localización, se propone un método basado en técnicas de SLAM. Concretamente, se implementa un SLAM basado en filtro extendido de Kalman (EKF) aplicado a secuencias de imágenes monoculares. La idea fundamental es aprovechar la información dada por la odometría basada en homografía para generar buenas hipótesis en la fase predictiva del filtro.

Las características naturales del entorno son utilizadas como puntos de referencia a seguir a lo largo de la secuencia de imágenes. La posición 3D de dichas marcas es estimada en el interior del filtro y utilizada para restringir el espacio de posibles movimientos del UAV.

Una de las principales aportaciones de este Capítulo es la inicialización de la posición de los puntos de referencia en el filtro de Kalman. Dado que la visión monocular solo aporta información de la escena en ángulo, pero no profundidad, la estimación de los puntos de referencia en 3D es un problema complejo. Para solucionarlo, el capítulo propone el uso del vector normal al plano de la escena y el sensor de rango, para así restringir la distancia a la que se encuentra el punto. Esta inicialización permite una rápida convergencia hacia la posición 3D, lo cual redunda en un mejor comportamiento del filtro.

B.9 Contribuciones de la Tesis

El objetivo de esta Tesis es el desarrollo de algoritmos de visión artificial para la localización de vehículos autónomos aéreos utilizando secuencias de imágenes monoculares. La visión monocular es una buena solución debido a que ofrece escalabilidad y precisión, unidos a un bajo peso. Sin embargo, la incertidumbre en la escala de las medidas tomadas por la cámara deber ser resuelta. Para este fin, la Tesis ha utilizado un sensor de rango que mide la distancia del vehículo a la escena con buenos resultados.

Un aspecto importante de la Tesis es el uso de rasgos naturales del entorno en lugar de balizas o referencia visuales con posiciones conocidas. En su lugar, se utiliza un algoritmo genérico de establecimiento y seguimiento de correspondencias entre imágenes. Aunque los rasgos naturales incrementan el nivel de aplicabilidad de los métodos propuestos en la tesis, también incrementan la complejidad del problema a ser resuelto. Así, se hace necesaria la implementación de algoritmos que detecten outliers y sistemas que garanticen una estimación robusta.

La Geometría Proyectiva es la teoría de fondo de la mayoría de los métodos propuestos. Las transformaciones homográficas se han usado para modelar el movimiento aparente entre imágenes consecutivas y como herramienta básica en muchos de los métodos propuestos. La Tesis describe un método robusto para la estimación de homografías, método usado en todos los experimentos presentados con muy buenos resultados. El estudio de las incertidumbres asociadas a dicha estimación también se ha tratado en la Tesis. Además, el método de estimación de homografías se ha extendido a escenas pseudo-planas mediante el desarrollo de un sistema jerarquizado de estimación.

La Tesis ha propuesto un sistema de odometría visual para UAVs basado en secuencias de imágenes monoculares y un sensor de rango. La descomposición de la homografía se ha utilizado para extraer el movimiento real del UAV y una estimación del vector normal al plano de la escena. Los resultados experimentales con UAVs reales muestran la viabilidad de la técnica, al menos con imágenes tomadas de 15 a 150 metros, y su eficiencia, hasta $10Hz$ con imágenes de resolución $320x240$.

El problema de la localización también se trata en esta Tesis. Se ha propuesto una técnica basada en la construcción de mosaicos en línea para reducir el error acumulativo presente en la odometría. Este método ha demostrado funcionar convenientemente cuando el UAV vuela a gran altura.

Además, se ha descrito una nueva técnica para la construcción de mosaicos que tiene en cuenta el carácter estocástico de las relaciones entre las imágenes componentes del mosaico. Esta técnica permite extender el uso de la localización basada en mosaicos a altitudes más bajas, donde el efecto de paralaje introduce distorsiones importantes, y aprovecha la información aportada por los cierres de bucle en posición para mejorar la estimación de las relaciones entre las imágenes que componen el mosaico.

La localización basada en mosaicos es especialmente adecuada para llevar a cabo tareas de monitorización o vigilancia con el UAV, debido a que se espera que el UAV visite repetidamente las mismas áreas. El marco estocástico permite mejorar, tanto las posiciones relativas de las imágenes cada vez que se cierra un bucle en posición, como la estimación de la localización.

Sin embargo, la localización basada en mosaicos no puede ser llevada a cabo cuando la distancia hasta el suelo cambia con frecuencia o la orientación de la cámara es significativamente oblicua con respecto al plano del suelo. Un método de SLAM basado en filtrado de Kalman se ha propuesto para abordar estas situaciones. La localización basada en homografía se utiliza para generar hipótesis de movimiento precisas. Además, se utiliza una nueva inicialización de puntos de referencia en el filtro de Kalman que aprovecha la estimación del vector normal al plano de la escena, permitiendo una buena convergencia hacia la posición real del punto de referencia y, por tanto, mejorando la convergencia del filtro completo.

B.10 Desarrollos futuros

B.10.1 Control del UAV basado en visión monocular

El primer objetivo del autor es aplicar en tiempo-real las técnicas de odometría y localización propuestas en esta Tesis. Estas técnicas se utilizarán para estimar el movimiento del helicóptero HERO en diversas situaciones tales como hovering, aterrizaje o despegue.

Estas aplicaciones aportarán información de posición sin el uso de dispositivos especiales de posicionamiento, lo cual es muy importante en escenarios donde han ocurrido desastres, de cualquier índole, debido a la necesidad de realizar un despliegue rápido y a que el GPS puede no estar disponible

B.10.2 Cálculo simultaneo de homografía y tracking

La reducción de la carga computacional necesaria para estimar el movimiento del UAV es otro objetivo del autor. Aunque el algoritmo presentado en el Capítulo 3 funciona correctamente, es necesario llevar a cabo seguimiento de correspondencias y cálculo de homografía antes de calcular el movimiento. Estos cálculos se podrían ver reducidos si la información de la homografía se pudiera utilizar para el cálculo de correspondencias y viceversa, es decir si se pudiera llevar a cabo un cálculo simultáneo de las correspondencias y de la homografía.

La homografía modela las correspondencias entre dos imágenes tomadas de la misma escena plana. Si un conjunto de correspondencias bien distribuidas por la imagen, al menos cuatro, fuesen proporcionadas por el algoritmo de seguimiento, se podría estimar una homografía. Aunque dicha estimación sería poco precisa debido al reducido número de correspondencias, probablemente aportaría una buena aproximación a la solución. Así, esta homografía podría ser utilizada para determinar las áreas de búsqueda para un subconjunto de nuevas correspondencias y, una vez calculadas, ser utilizadas para re-estimar la homografía. Cuantas más correspondencias se utilicen para re-estimar la homografía, mejor será el resultado y más precisas serán las áreas

de búsqueda de correspondencias, lo cual conduce a una reducción en los tiempos de cómputo.

Además, si se tiene en cuenta la identidad (3.9), la homografía puede ser reemplazada por una expresión que es función de la traslación, la rotación y la normal al plano. Esto daría lugar a un método iterativo sencillo para el cálculo simultáneo del seguimiento de correspondencias y del movimiento real del vehículo.

B.10.3 Inspección de edificios

La estimación relativa de posición con respecto a una pared es una aplicación interesante de la odometría basada en homografía. La estimación del GPS suele ser imprecisa cuando el UAV vuela cerca de edificios. La odometría basada en homografía puede ser una importante herramienta para llevar a cabo maniobras cercanas a edificios o interpretar datos procedentes de paredes.

Así, la odometría basada en homografía es capaz de obtener la orientación relativa de la cámara con respecto a la pared. Esta información puede ser utilizada para controlar el helicóptero con el fin de obtener una visión perpendicular de la pared inspeccionada. Dada la distancia inicial entre la cámara y la pared, el algoritmo también puede ser capaz de recuperar la distancia a la pared cada vez que se reciba una imagen.

Este algoritmo puede ser utilizado incluso para generar vistas ortogonales virtuales de la pared inspeccionada, dado que se estima el vector normal al plano. La vista ortogonal puede llegar a ser una herramienta muy importante cuando es necesario reconocer características geométricas de la vista.

B.10.4 Navegación basada en mosaicos geo-referenciados

Se investigará el uso de mosaicos geo-referenciados en la localización basada en mosaicos descrita en el Capítulo 4. La idea básica es usar mosaicos, previamente calculados y con referencias geográficas, para lograr una localización global del vehículo.

Las imágenes capturadas por la cámara montada en el vehículo serán comparadas con el mosaico para calcular su posición exacta. Después, si el mosaico está geo-referenciado, la relación de la imagen con el mosaico puede ser utilizada para calcular la posición real del UAV. Esta aplicación es realmente interesante cuando el UAV se desplaza siempre por la misma zona.

References

Amidi, O., Kanade, T., and Fujita, K. (1998). A visual odometer for autonomous helicopter fight. In *Proceedings of the Fifth International Conference on Intelligent Autonomous Systems (IAS-5)*.

Betge-Brezetz, S., Hebert, P., Chatila, R., and Devy, M. (1996). Uncertain map making in natural environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1048–1053.

Betke, M. and Gurvits, L. (1997). Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13:251–263.

Bosch, S., Lacroix, S., and Caballero, F. (2006). Autonomous detection of safe landing areas for an uav from monocular images. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006*, pages 5522–5527, Beijing, China.

Byrne, J., Cosgrove, M., and Mehra, R. (2006). Stereo based obstacle detection for an unmanned air vehicle. In *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pages 2830–2835.

Caballero, F., Merino, L., Ferruz, J., and Ollero, A. (2005). A visual odometer without 3d reconstruction for aerial vehicles. applications to building inspection. In *Proceedings of the International Conference on Robotics and Automation*, pages 4684–4689. IEEE.

Caballero, F., Merino, L., Ferruz, J., and Ollero, A. (2006). Improving vision-based planar motion estimation for unmanned aerial vehicles through online mosaicing. In *Proceedings of the International Conference on Robotics and Automation*, pages 2860–2865. IEEE.

Caballero, F., Merino, L., Ferruz, J., and Ollero, A. (2007). Homography based kalman filter for mosaic building. applications to uav position estimation. In *IEEE International Conference on Robotics and Automation, 2007*, pages 2004–2009.

Davison, A. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *IEEE International Conference on Computer Vision*, pages 1403–1410.

Deans, M. and Hebert, M. (2000). Experimental comparison of techniques for localization and mapping using a bearings only sensor. In *Proceedings of the Seventh International Symposium on Experimental Robotics*.

Demonceaux, C., Vasseur, P., and Pegard, C. (2006). Omnidirectional vision on uav for attitude computation. In *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pages 2842–2847.

Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images: The laws that govern the formation of multiple images of a scene and some of their applications*. The MIT Press.

Feder, H., Leonard, J., and Smith, C. (1999). Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18(7):650–668.

Ferruz, J. and Ollero, A. (2000). Real-time feature matching in image sequences for non-structured environments. applications to vehicle guidance. *Journal of Intelligent and Robotic Systems*, 28:85–123.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Garcia, R., Batlle, J., Cufí, X., and Amat, J. (2001). Positioning an underwater vehicle through image mosaicking. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, volume 3, pages 2779–2784, Seoul, Rep. of Korea.

Garcia, R., Puig, J., Ridao, P., and Cufi, X. (2002). Augmented state kalman filtering for auv navigation. In *Proceedings of the International Conference on Robotics and Automation*, pages 4010–4015, Washington.

Garcia-Pardo, P. J., Sukhatme, G. S., and Montgomery, J. F. (2001). Towards vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous Systems*, 38(1):19–29.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

Hrabar, S. and Sukhatme, G. (2003). Omnidirectional vision for an autonomous helicopter. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 558–563.

Hsu, S., Sawhney, H. S., and Kumar, R. (2002). Automated mosaics via topology inference. *IEEE Comput. Graph. Appl.*, 22(2):44–54.

Huber, P. (1981). *Robust Statistics*. Wiley.

Hygounenc, E., Jung, I.-K., Soueres, P., and Lacroix, S. (2004). The Autonomous Blimp Project of LAAS-CNRS: Achievements in Flight Control and Terrain Mapping. *The International Journal of Robotics Research*, 23(4-5):473–511.

Johnson, E. and Schrage, D. (2003). The georgia tech unmanned aerial research vehicle: Gtmax. In *Proceedings of the 2004 AIAA Guidance, Navigation, and Control Conference*.

Julier, S. and Uhlmann, J. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45.

Kim, J. and Sukkarieh, S. (2004). Autonomous airborne navigation in unknown terrain environments. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1031–1045.

Kwok, N. M. and Dissanayake, G. (2004). An efficient multiple hypothesis filter for bearing-only slam. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 736–741.

Lacroix, S., Jung, I., Soueres, P., Hygounenc, E., and Berry, J. (2002). The autonomous blimp project of LAAS/CNRS - current status and research challenges. In *Proceeding of the International Conference on Intelligent Robots and Systems, IROS, Workshop WS6 Aerial Robotics*, pages 35–42. IEEE/RSJ.

Leonard, J. and Durrant-Whyte, H. (1991). Simultaneous map building and localization for an autonomous mobilerobot. In *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1442–1447.

Mahalanobis, P. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Science of India*, 12:49–55.

Mahony, R. and Hamel, T. (2005). Image-based visual servo control of aerial robotic systems using linear image features. *IEEE Transactions on Robotics*, 21(2):227–239.

Martinez, J. R., Merino, L., Caballero, F., Ollero, A., and Viegas, D. X. (2006). Experimental results of automatic fire detection and monitoring with uavs. *Forest Ecology and Management*, 234S:S232.

Mejías, L., Saripalli, S., Campoy, P., and Sukhatme, G. S. (2006). Visual servoing of an autonomous helicopter in urban areas using feature tracking. *Journal of Field Robotics*, 23(3-4):185–199.

Merino, L., Caballero, F., Martinez, J. R., Ferruz, J., and Ollero, A. (2006a). A cooperative perception system for multiple uavs: Application to automatic detection of forest fires. *Journal of Field Robotics*, 23(3):165–184.

Merino, L., Wiklund, J., Caballero, F., Moe, A., de Dios, J. R. M., Forssén, P.-E., Nordberg, K., and Ollero, A. (2006b). Vision-based multi-uav position estimation. *IEEE Robotics and Automation Magazine*, 13(3):53–62.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada. AAAI.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico. IJCAI.

Montiel, J., J, C., and Davison, A. (2006). Unified Inverse Depth Parametrization for Monocular SLAM. In *Robotics: Science and Systems*.

Ollero, A., Ferruz, J., Caballero, F., Hurtado, S., and Merino, L. (2004). Motion compensation and object detection for autonomous helicopter visual navigation in the comets system. In *Proceedings of the International Conference on Robotics and Automation, ICRA*, pages 19–24. IEEE.

Ollero, A., Lacroix, S., Merino, L., Gancet, J., Wiklund, J., Remuss, V., Veiga, I., Gutierrez, L. G., Viegas, D. X., A.Gonzalez, M., Mallet, A., Alami, R., Chatila, R., Hommel, G., Colmenero, F. J., Arrue, B., Ferruz, J., Martinez, J. R., and Caballero, F. (2005). Multiple eyes in the sky: Architecture and perception issues in the COMETS unmanned air vehicles project. *IEEE Robotics and Automation Magazine*, 12(2):46–57.

Papadopoulo, T. and Lourakis, M. (2000). Estimating the jacobian of the singular value decomposition: Theory and applications. In *Proceedings of the 2000 European Conference on Computer Vision*, volume 1, pages 554–570.

Pari, L., Sebastian, J. M., Gonzalez, C., and Angel, L. (2006). Image based visual servoing: A new method for the estimation of the image jacobian in dynamic environments. *Lecture Notes in Computer Science*, 4141:850–861.

Proctor, A., Johnson, E., and Apker, T. (2006). Vision-only control and guidance for aircraft. *Journal of Field Robotics*, 23(10):863–890.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. Wiley-IEEE.

Rzhanov, Y., Huff, L., and Cutter, R. (2001). Improvement of image alignment using camera attitude information. In *Proceedings of the Sixth International Symposium on Signal Processing and its Applications*, volume 2, pages 639–642, Kuala Lumpur, Malaysia.

Saripalli, S., Montgomery, J., and Sukhatme, G. (2003). Visually guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–380.

Saripalli, S. and Sukhatme, G. (2003). Landing on a mobile target using an autonomous helicopter. In *Proceedings of the International Conference on Field and Service Robotics, FSR*.

Schultz, H., Hanson, A. R., Riseman, E. M., and Stolle, F. (2002). Rapid updates of gis databases from digital images. In *Proceedings of the 2002 annual national conference on Digital government research*, pages 1–7. Digital Government Research Center.

Shakernia, O., Vidal, R., Sharp, C., Ma, Y., and Sastry, S. (2002). Multiple view motion estimation and control for landing an aerial vehicle. In *Proceedings of the*

International Conference on Robotics and Automation, ICRA, volume 3, pages 2793–2798. IEEE.

Shum, H.-Y. and Szeliski, R. (1998). Construction and refinement of panoramic mosaics with global and local alignment. In *Proceedings of the Sixth International Conference on Computer Vision, ICCV*, page 953, Washington, DC, USA. IEEE Computer Society.

Smith, R. and Cheeseman, P. (1987). On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, pages 56–68.

Tardos, J., Neira, J., Newman, P., and Leonard, J. (2002). Robust mapping and localization in indoor environments using sonar data. *Int. J. of Robotics Research*, 24(4):311–330.

Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., and Durrant-Whyte, H. (2004). Simultaneous localization and mapping with sparse extended information filters. *Internation Journal of Robotics Research*, 23(7–8):693–716.

Triggs, B. (1998). Autocalibration from planar scenes. In *Proceedings of the 5th European Conference on Computer Vision, ECCV*, volume 1, pages 89–105, London, UK. Springer-Verlag.

Tsai, R., Huang, T., and Zhu, W.-L. (1982). Estimating three-dimensional motion parameters of a rigid planar patch, ii: singular value decomposition. *IEEE Transaction Acoustic, Speech and Signal Processing*, 30(8):525–524.

Vidal, R., Sastry, S., Kim, J., Shakernia, O., and Shim, D. (2002). The berkeley aerial robot project (bear). In *Proceeding of the International Conference on Intelligent Robots and Systems, IROS*, pages 1–10. IEEE/RSJ.

Vidal-Calleja, T., Bryson, M., Sukkarieh, S., Sanfeliu, A., and Andrade-Cetto, J. (2007). On the observability of bearing-only slam. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pages 1050–4729.

Volpe, J. A. (2001). Vulnerability assessment of the transportation infrastructure relying on the global positioning system. Technical report, Office of the Assistant Secretary for Transportation Policy.

Welch, G. and Bishop, G. (1995). An introduction to the kalman filter. Technical Report 95-041, University of North Carolina at Chapel Hill.

Zhang, Z. (1995). Parameters estimation techniques. a tutorial with application to conic fitting. Technical report, INRIA, France.

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the 7th International Conference on Computer Vision, ICCV*, pages 666–673, Greece.