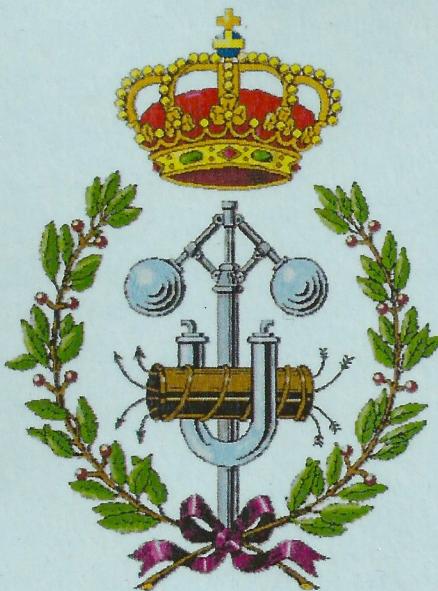


UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS INDUSTRIALES



*On-board visual control algorithms for
Unmanned Aerial Vehicles*

A thesis submitted for the degree of
Doctor of philosophy in Robotics

Iván Fernando Mondragón Bernal
BSEE Electrical Engineer

2011



Universidad Politécnica de Madrid
Escuela Técnica Superior De Ingenieros Industriales
Doctorado en Automática y Robótica

**On-board visual control algorithms for Unmanned
Aerial Vehicles.**

**A thesis submitted for the degree of *Doctor of
philosophy in Robotics***

Iván Fernando Mondragón Bernal
BSEE Electrical Engineer

2011



Escuela Técnica Superior De Ingenieros Industriales
Doctorado en Automática y Robótica
Universidad Politécnica de Madrid

On-board visual control algorithms for Unmanned Aerial Vehicles.

A thesis submitted for the degree of *Doctor of
philosophy in Robotics*

Author:
Iván Fernando Mondragón Bernal
BSEE Electrical Engineer

Director:
Dr. Pascual Campoy Cervera
PhD Full Professor Industrial Engineer

2011

Título:
**On-board visual control algorithms for Unmanned
Aerial Vehicles.**

Autor:
Iván Fernando Mondragón Bernal
BSEE Electrical Engineer

Tribunal nombrado por el Mgfco. y Excmo. Sr Rector de la Universidad
Politécnica de Madrid el día.....de.....de 2011

Tribunal

Presidente :,

Secretario :,

Vocal :,

Vocal :,

Vocal :,

Suplente :,

Suplente :,

Realizado el acto de lectura y defensa de la tesis el día de de 2011.

calificación de la Tesis.....

El Presidente:

Los Vocales:

El Secretario:

A mis Padres Eduardo y Alicia.

A mi tía Stella.

A mis hermanos Nohora, Olga, Nelson y Javier.

A sus esposas y esposos Juan, Rafael, Jose Guillermo, Swinda y Eliana, y a mis sobrinas Natalia, Sofia y Mariana.

A mi querida esposa Claudia Cristina

A toda mi familia y a los futuros miembros de esta.

A ellos estas palabras por brindarme su cariño, confianza y por sus enormes sacrificios y esfuerzos, además de ser y dar la fuerza que te motiva a seguir adelante.

Iván Fernando.

Acknowledgements

It is a pleasure to thank those who made this thesis possible. First of all, I am heartily thankful to my supervisor, professor Pascual Campoy Cervera, whose encouragement, guidance and support given during the last five years.

I would like to show my gratitude for all the past and current members of the Computer Vision Group, among them, must be mentioned, the professors José María Sebastian and Sergio Dominguez, Carol Martínez, Miguel Olivares, Juan Carlos Garcia, Ignacio Mellado, David Galindo, José Luís Sanchez.

Thanks to all the DISAM members, from whom I received an invaluable collaboration and support: Specially thanks to Teresa, Rosa, Carlos, Angel, Jaime, Enrique and the professors Rafael Aracil, Agustín Jiménez, Ramón Galán, Antonio Barrientos, Manuel Ferre, Claudio Rossi, Roque Saltarén, Ernesto Gambao, Ricardo Sanz and all others members of this great laboratory.

I would like to thank institutions like the Consejería de Educación de la Comunidad de Madrid and the Fondo Social Europeo FSE for the Author PhD scholarship through the program “Personal Investigador de Apoyo PIA, 2008” and the Ministerio de Educación y Ciencia for grant support in the project DPI2007-66156 “Visión por computador para UAV: Navegación, seguimiento e inspección”. I’m also grateful with the Universidad Politécnica de Madrid for the financial support for the European stay as well as the European Community IRSES FP7 program for the collaboration on the ARCAA research stay.

I also want to thank to the Autonomous Systems Laboratory LSA at Instituto Superior de Engenharia do Porto ISEP - (Portugal) and The Australian Research Centre for Aerospace Automation ARCAA - (Australia), and specially to professors Eduardo Silva and Luis Mejias, for give me the opportunity to visit, collaborate and gain new experiences in both the personal and the researching level. I also want to express my gratitude with professor Patrick Doherty and Piotr Rudol for the suggestions an effort made on reviewing this work

I'm very grateful with all my family; my wife, my parents, my aunt, my sisters and brothers. I'll be forever grateful with all of them for the support and confidence that they have give me.

Finally, thanks to all the persons that have collaborated on the development of this thesis and have not been mentioned.

Resumen

El objetivo de esta tesis, es el de utilizar la gran cantidad de información que los sistemas de visión proveen, con el objetivo de diseñar sistemas de guiado visual para vehículos aéreos no tripulados (UAVs por sus siglas en Ingles). Este trabajo va más allá del uso tradicional de sistemas visuales en UAVs, de ser simples "ojos en el cielo", demostrando que un sensor visual provee suficiente información confiable y robusta, que puede ser empleada para el diseño de diferentes aplicaciones y sistemas de control, manteniendo una precisión y desempeño similar o en algunos casos mayor a los demandados en otros sensores de control usualmente empleados en este tipo de sistemas.

Este trabajo, presenta en primera medida, como algunos de los algoritmos pioneros para el procesamiento de imágenes, son robusta y eficientemente implementados a bordo de Vehículos Aéreos no tripulados y luego utilizados para la estimación de diversas transformaciones proyectivas. Estas transformaciones (incluyendo las homografías), son empleadas para el procesamiento y mejoramiento de imágenes aéreas, presentando en primera medida un sistema de estabilización de video en tiempo real al igual que un método para la generación de mosaicos. Por otra parte, las propiedades de las transformaciones proyectivas, son empleadas para el desarrollo de un nuevo sistema de medición de posición en el espacio tridimensional, permitiendo obtener la información métrica de la posición al igual que la orientación de la cámara (y el vehículo aéreo) con respecto a un plano de referencia.

La información métrica de la posición y orientación obtenida mediante la descomposición de la homografía, es empleada para generar un sistema de control de alto nivel del tipo desacoplado y que es basado en la posición, el cual presenta un excelente desempeño ante la dinámica no holonómica de un vehículo aéreo no tripulado del tipo ala rotatoria, al igual que una gran robustez con respecto a los problemas comúnmente presentes en imágenes aéreas, como lo son las altas vibraciones y los constantes cambios de iluminación, al igual que los cambios de posición o las occlusiones de las referencias visuales empleadas, entre otros. Esta contribución es luego empleada para el control de un UAV en tareas como el aterrizaje autónomo y posicionamiento estático de precisión.

Las estrategias de control desacopladas, también son empleadas para el diseño de un controlador visual basado en la imagen, en un novedoso sistemas empleado para el seguimiento de objetos aéreos móviles. En este sistema, el análisis de la interacción entre las referencias visuales y los grados de libertad del sistema es empleada para reducir los complejidad del diseño y ajuste del controlador visual dadas por la dinámica del vehículo aéreo.

Esta tesis también presenta como los sistemas de visión omnidireccional son empleados para estimar la actitud de vuelo del vehículo aéreo al igual que para el diseño de un compás visual. El método propuesto, se basa en las propiedades de la teoría unificada para cámaras del tipo catadióptrico, es empleado para generar un sensor de actitud de vuelo basado en la segmentación de la línea del horizonte y del movimiento entre imágenes consecutivas.

Finalmente, las propiedades de los sistemas de control del alto nivel y del tipo desacoplados, al igual que de los sistemas de visión omnidireccional son integradas con el animo de genera un novedoso método para la evasión de colisiones aéreas. Esta contribución en el área de "Observar y Evadir" emplea una arquitectura de control visual desacoplado, diseñada directamente en el espacio de trabajo de la esfera unitaria, permitiendo integrar las propiedades y ventajas de el gran ángulo de visión de los sistemas de visión omnidireccional en el diseño de sistemas de control visual para Vehículos Aéreos no tripulados.

Los diversos métodos propuestos han sido probados y validados en vuelos reales utilizando vehículos aéreos no tripulados del tipo ala rotatoria. Sin embargo las contribuciones presentadas en esta tesis son fácilmente extendibles a otros tipos de Vehículos aéreos al igual que a la robótica móvil en general.

Esta tesis demuestra que la visión por computador abordo de UAV es un campo sin explotar. Esta ha demostrado ser un sensor de gran versatilidad, bajo costo y alta efectividad para muchas aplicaciones industriales al igual que ser un componente fundamental para la operación segura de vehículos aéreos no tripulados. Finalmente, esta tesis abre las puertas al desarrollo de nuevos sistemas para su uso en la industria y en el creciente mercado del sector civil, en el cual es fundamental el uso de información visual que permita incrementar las autonomía y prestaciones al igual que garantizar la seguridad de operación de este tipo de aeronaves.

Abstract

The aim of this thesis is to exploit the vast amount of information given by a visual sensor in order to design visually guided Unmanned Aerial Vehicles. This work, goes beyond the traditional use of visual systems on a UAV, of being a simple "eye in the sky", by demonstrating that a visual sensor provides trusted and reliable information useful for designing different control systems and applications with the capacity, precision and performance similar to the ones demanded in traditional control sensors.

This thesis presents the form in which the state of the art of visual tracking and image processing algorithms are robustly and efficiently implemented on-board UAVs in order to estimate different projective transformations. These transformations (homographies included), are then used for aerial image enhancement, presenting real time video stabilization and a mosaic building system. The properties of the projective transformation are then employed to build a 3D pose measuring system that allows obtaining the camera-aircraft metric position and attitude w.r.t. as ground reference plane.

The metric and attitude positions estimated by the homography decomposition, are used to generate a high level decoupled pose control system that correctly drives the non-holonomic behavior of a rotary wing UAV and the common challenges presented on aerial images, such as higher vibrations, changes in illumination, visual reference occlusions and large displacements, among others. The presented contribution is used for controlling UAVs in situations like autonomous landing and accurate positioning.

High Level decoupled control strategies are also used for developing an Image-based visual servoing architecture for a novel aerial moving object following system, in which the analysis of the interaction between the image references and the system degrees of freedom (DOF) are used in order to reduce the effects of the aircraft dynamics on the visual control design and tuning.

This thesis also presents how Omnidirectional systems are used for estimat-

ing the aircraft flying attitude and making visual compass. The proposed method uses the properties of unified theory for central catadioptric cameras in order to build an attitude sensor based on skyline segmentation and motion registration between sequential omnidirectional images.

Finally, the properties of a decoupled visual control and an omnidirectional system are integrated in order to generate a novel collision avoidance system. This contribution in the field of See&Avoid area, uses a decoupled Image-based visual architecture defined on the unitary sphere manifold that allows including the properties and advantages of a large field of view (FOV) given by a omnidirectional sensor in the design of a visual control systems for UAV.

All the proposed methods are tested and validated in real aerial tests using rotary wing UAVs. The contributions presented in this thesis can be easily extended to other types of UAV as well as general mobile robotics.

This thesis demonstrates that computer vision on-board UAV is a versatile, low cost and effective sensor systems suitable for many industrial applications as well as a principal component for UAV safety operation. This thesis opens the door to the development of novel systems for industrial and civilian markets in which there is necessity for using visual data in order to increase aircraft autonomy and capabilities that guarantee safety in several operational levels.

Nomenclature

The conventions used in this document are defined as follows:

- Matrix will be represented by capital bold letters such as **H** or **R**.
- Vectors will be represented by a lower case bold letters with or without a upper bar, such as \bar{x} or **t**.
- A vector is a column vector, unless indicated otherwise.
- Scalar variables are normal italic letters such as *i* or *k*.
- Images are normally represented by capital italic letters such as *I* or *T*.

Acronyms

Following is a description of most commonly acronyms referenced on this work:

ATC: Air Traffic Control

ATM: Air Traffic Management

CAA: United Kingdom Civil Aviation Authority

CASA: Australian Civil Aviation Safety Authority

CPU: Central Processing Unit

DOF: Degrees of freedom

EASA: European Aviation Safety Agency

FAA: Federal Aviation Administration

FOV: Field Of View

GPS: Global Positioning System

GPU: Graphics Processing Unit

IBVS: Image-Based Visual Servoing

ICIA: Inverse Compositional Image Alignment

HOT: Hight Order Terms

IMU: Inertial Measurement Unit

KF: Kalman Filter

LKT: Lucas-Kanade Tracker

MAV: Micro-Aerial Vehicles.

MTOW: Maximum Take-Off Weight

NAAs: National Aviation Authorities

NAS: National Airspace System

PBVS: Position-Based Visual Servoing

PID: : Proportional-Integral-Derivative controller

RANSAC: RANdom SAmple Consensus

RC: Radio Controlled

RMSE: Root Mean Square Error

RPM: Revolutions per Minute

R&D: Research and Development

SIFT: Scale-Invariant Feature Transform

SLAM: Simultaneous Localization And Mapping

SURF: Speeded Up Robust Features

UAS: Unmanned Aircraft System

UAV: Unmanned Aerial Vehicle

VTOL: Vertical Take-Off and Landing

Contents

Acknowledgements	VII
Resumen	IX
Abstract	XI
Nomenclature	XIII
Acronyms	XV
List of Figures	XXI
List of Tables	XXV
1. Introduction	1
1.1. Challenges for computer vision and UAVs	1
1.2. Problem Statement	7
1.3. Thesis proposal and objectives	9
1.4. Thesis Contributions	9
1.5. Dissertation outline	11
2. State of the Art	13
2.1. Introduction	13
2.2. Computer Vision for UAV	14
3. Image Processing for UAV applications	21
3.1. Introduction	21
3.2. Image Based Visual Tracking	21

3.2.1. Appearance tracking	22
3.2.2. Feature descriptors and tracking	26
3.2.3. Robust matching	28
3.3. 3D estimation based on Homographies	32
3.3.1. World plane projection onto the Image plane	33
3.4. 3D pose performance evaluation	36
3.4.1. 3D pose tests and results	37
3.5. Conclusions	42
4. UAVs on-board visual odometry and video enhancement	43
4.1. Introduction	43
4.2. Aerial Video Stabilization	43
4.3. On-board mosaic building	47
4.4. UAV 3D pose estimation based on planar landmarks	53
4.4.1. 3D Pose data filtering	59
4.4.2. UAV 3D pose tests and results	61
4.5. Conclusions	65
5. Visual Control for rotary wing UAV	67
5.1. Introduction	67
5.2. Visual Servoing	68
5.2.1. Camera Configurations	68
5.2.2. Visual control schemes	70
5.2.3. Visual control schemes according to the definition of the error function	72
5.3. Landing and Hovering based on visual pose control	77
5.3.1. Tests and results	79
5.4. Aerial object following based on visual information	87
5.4.1. Color-based probabilistic tracking approach	91
5.4.2. Tests and results	92
5.5. Conclusions	98
6. Omnidirectional Vision for UAVs	101
6.1. Introduction	101
6.2. Central Catadioptric Cameras and Omnidirectional Images . . .	102
6.2.1. Mirror Design	106
6.2.2. Fisheye cameras and the unify model	110
6.3. UAV attitude estimation based on omnidirectional image processing	110
6.3.1. Skyline and catadioptric Image.	113
6.3.2. Skyline backprojection on sphere.	114
6.3.3. <i>Pitch</i> and <i>Roll</i> estimation	117
6.3.4. <i>Yaw</i> estimation using a visual compass	117
6.3.5. Tests and Results	119
6.4. UAVs See&Avoid	125

6.4.1. Detection Approach	127
6.4.2. Approach Implementation	129
6.4.3. Tests and results using quadrotors	131
6.5. Conclusions	133
7. Conclusions and Future Work	137
7.1. Conclusions	137
7.2. Future Work	140
A. Publications derived from this thesis	143
A.1. Journals with JCR	143
A.2. Congress	144
A.3. Book Chapters	146
A.4. Digital media	146
B. UAV specifications	147
B.1. Gas and Electric powered UAV helicopters	147
B.1.1. Helicopter Autopilot API	147
B.2. UAV quadcopter	151
B.2.1. Quadcopter autopilot SDK	151
C. Research Groups on UAS and Vision Systems	155

List of Figures

1.1.	The European Civilian UAS Market 2008-2017: Phasing-in of a regulatory framework.	5
1.2.	The European civilian UAS market 2008-2017 according to the industrial application.	6
2.1.	Vision on UAVs generic scheme	20
3.1.	Comparison between features point extractors.	29
3.2.	Robust homography estimation using SURF features on a car tracking from a UAV.	33
3.3.	Projection model on a moving camera and frame-to-frame homography induced by a plane	34
3.4.	Vicon camera model	37
3.5.	Vicon work space	38
3.6.	3D pose estimation based on landmarks performance test	38
3.7.	2D trajectory and camera heading reconstruction	39
3.8.	Comparison between the <i>X</i> axis displacement for homography estimation and VICON reference System.	40
3.9.	Comparison between the <i>Y</i> axis displacement for homography estimation and VICON reference System.	40
3.10.	Comparison between the <i>Z</i> axis displacement for homography estimation and VICON reference System.	41
3.11.	Comparison between the <i>Yaw</i> rotation for homography estimation and VICON reference System.	41
4.1.	Optical flow between consecutive images using the pyramidal Lucas Kanade Tracker.	44

4.2.	Video Stabilization of image I_{k+1} with respect to the reference image on the sequence.	46
4.3.	Real time Video Stabilization on a UAV sequence of 700 images (1024x780 pixels) using a GPU LKT implementation	48
4.4.	Real time Video Stabilization on a sequence of 500 images using a GPU LKT implementation	49
4.5.	Mosaic Building process	51
4.6.	Mosaic Building based on 450 images taken from a UAV	52
4.7.	Mosaic Building based on 620 images taken with a hand held camera	54
4.8.	Mosaic Building based on 300 images taken with the Colibri 3 UAV	55
4.9.	Helipad used as a plane reference for UAV 3D pose estimation based on homographies.	56
4.10.	Helipad, camera and U.A.V coordinate systems.	57
4.11.	Homography motion model estimated using either, the Lucas Kanade Algorithm with RANSAC robust function fitting or with the Inverse Compositional Algorithm ICIA.	57
4.12.	Two different test for 3D pose estimation based on a helipad tracking using robust homography estimation.	60
4.13.	3D flight and heading reconstruction. Superimposed images show the helipad at different times of the test, from which the 3D position is estimated.	61
4.14.	Comparison between the X axis displacement for homography estimation and IMU+GPS state estimated data.	62
4.15.	Comparison between the Y axis displacement for homography estimation and IMU+GPS state estimated data.	63
4.16.	Comparison between the Z axis displacement for homography estimation and IMU+GPS state estimated data.	63
4.17.	Comparison between the Yaw angle measured using homography estimation and IMU+GPS state estimated data.	64
4.18.	3D pose estimation occlusion robustness.	64
5.1.	Camera configuration for UAV visual servoing.	69
5.2.	Direct Visual Servoing.	71
5.3.	Dynamic look and move control.	72
5.4.	UAV control <i>Dynamic pose-based look-and-move</i> system architecture.	78
5.5.	Using a helipad as a plane reference for UAV 3D pose estimation based on homographies	80
5.6.	3D reconstruction of the flight test using the IMU+GPS values and the vision-based estimation.	81
5.7.	On-board UAV visual pose control	82
5.8.	Helipad position on each of the three different landing tests . . .	82

5.9. Landing process using a 3D pose estimation based on a helipad tracking	83
5.10. Comparison between the X axis displacement for homography estimation and IMU+GPS data. Average frame rate is 12 Hz	84
5.11. Comparison between the Y axis displacement for homography estimation and IMU+GPS data. Average frame rate is 12 Hz	84
5.12. Comparison between the Z axis displacement for homography estimation and IMU+GPS data. Average frame rate is 12 Hz	85
5.13. Comparison between the Yaw angle measured using homography estimation and IMU+GPS data. Average frame rate is 12 Hz	85
5.14. UAV object following setup in 2D	88
5.15. Camshift tracking of a colored red target on an image sequence.	91
5.16. Object following <i>Dynamic image-based look-and-move</i> system architecture.	92
5.17. Two different object following tests using an Image-based visual servoing IBVS on a Pelican quadrotor.	93
5.18. $\widehat{\mathbf{L}}_e^+$ IBVS step response on u^* change.	94
5.19. $\widehat{\mathbf{L}}_e^+$ IBVS step response on ϕ^* change.	95
5.20. 2D reconstruction for test with steps changes on u^* and ϕ^* presented in Figures 5.18(d) and 5.19b	95
5.21. 2D reconstruction for Object following test	96
5.22. $\widehat{\mathbf{L}}_e^+$ IBVS response on u^* for a moving target following test.	97
5.23. $\widehat{\mathbf{L}}_e^+$ IBVS response on ϕ^* for a moving target following test.	97
5.24. $\widehat{\mathbf{L}}_e^+$ IBVS Yaw for a moving target following test.	98
6.1. Four catadioptric systems that satisfied the single effective viewpoint constrain.	103
6.2. Two degenerated configurations combining either, a spherical or conical mirror and a perspective camera.	104
6.3. Gayer Unitary Sphere Projection Model to the plane	104
6.4. Catadioptric projection modeled by the unit sphere.	105
6.5. Svoboda <i>et al.</i> model for a perspective camera with a hyperbolic Mirror.	107
6.6. Catadioptric system design and assembly using a transparent extruded thermoplastic acrylic tube.	110
6.7. The unify model on Fisheye images	111
6.8. Skyline projection on the unitary sphere model.	113
6.9. Skyline segmentation on a catadioptric image	114
6.10. Examples of the best fitted ellipse of the skyline.	115
6.11. Attitude estimation based on the skyline projection on the unitary sphere.	116
6.12. Two consecutive appearance images with a small rotation between them	118

6.13. Euclidean distance between the two appearance images.	118
6.14. Flight number 2, 3D trajectory and UAV heading reconstruction, obtained using the flightlog data.	120
6.15. 2D trajectories and UAV heading reconstruction.	121
6.16. Estimated <i>Roll</i> compared with the <i>Roll</i> angle measured by IMU	123
6.17. Estimated <i>Pitch</i> compared with the <i>Pitch</i> angle measured by IMU	124
6.18. Estimated <i>Relative Yaw</i> compared with the <i>Relative Yaw Angle</i> measured by IMU	125
6.19. Owncraft-Target See&Avoid setup in 2D.	126
6.20. Angle estimation on the own-craft.	128
6.21. 2D trajectories and UAV heading reconstruction for See&Avoid Test	132
6.22. UAV to Target Distance for See&Avoid Test	133
6.23. Estimated relative target <i>Yaw</i> angle compared with the <i>Relative</i> <i>Yaw</i> angle measured by IMU and the command <i>Yaw</i> Generated .	134
B.1. UAVs rotary wing helicopters used on this thesis	148
B.2. UAVs rotary wing quadcopter used on this thesis	152

List of Tables

1.1.	UAV classification according to mass, range, flight altitude and endurance	2
3.1.	3D pose performance tests	39
6.1.	Parameters ξ and ψ for central catadioptric systems.	106
6.2.	Flight test description and weather conditions.	120
6.3.	RMSE for estimated <i>Roll</i> (ϕ), <i>Pitch</i> (θ) and <i>Relative Yaw</i> (ψ). .	122
B.1.	Rotomotion UAVs technical specifications	149
B.2.	Pelican AscTec UAV technical specifications	153

Chapter 1

Introduction

1.1. Challenges for computer vision and UAVs

An Unmanned Aerial Vehicle UAV is defined as a pilotless aircraft which is flown without a pilot in command on-board and is either remotely and fully controlled from another place (ground, another aircraft, space) or programmed and fully autonomous. UAVs are often part of a more general structure designated as a Unmanned Aircraft System UAS, that in addition to the air vehicle, includes all needed hardware, structures and human resources required for operating a UAV. These include the sensors and payloads, command and control data links, the operator station, the ground support team and equipment necessary to launch and recover, operations and maintenance, among others. UAVs have been an active research area for many years and become attractive for civilian applications. It is due to its capability to carry out what is termed as D³ tasks (Dull, Dangerous and Dirty) (Welch and Bishop, 2005). It allows them to stay at remote locations for extended periods and in dangerous situations. When combined with an adequate payload, UAVs are a relative cheap and efficient aerial sensing platform since they offer very effective on-site data acquisition systems. They also provide high sensitivity and accurate measurements for a number of functions, like border control, fire fighting, ground traffic surveillance, and pollution control, among others.

There are many classifications for UAVs based on a variety of parameters that include mass, vehicle configuration, designed application, level of autonomy, type of operation, lift generation method, take-off method and control type, among others. However, there is not a unique consensus on the categorization. One of the most comprehensive and accepted classification has been proposed by Blyenburgh (Blyenburgh, 2006) (Table 1.1) which is based on weight, endurance and operational altitude.

Another common classification method often used by the National Aviation

Table 1.1: UAV classification according to mass, range, flight altitude and endurance

	Mass (kg)	Range (km)	Flight Alt. (m)	Endurance (h)
Micro	<5	<10	250	1
Mini	<20/25/30/150 ¹	<10	150/250/300	<2
Tactical				
Close range (CR)	25-150	10-30	3.000	2-4
Short range (SR)	50-250	30-70	3.000	3-6
Medium range (MR)	150-500	70-200	5.000	6-10
MR endurance (MRE)	500-1500	>500	8.000	10-18
Low altitude deep penetration (LADP)	250-2500	>250	50-9.000	0.5-1
Low altitude long endurance (LALE)	15-25	>500	3.000	>24
Medium altitude long endurance (MALE)	1000-1500	>500	3.000	24-48
Strategic				
High altitude long endurance (HALE)	2500-5000	>2.000	20.000	24-48
Stratospheric (Strato)	>2.500	>2.000	>20.000	>48
Ex-stratospheric (EXO)	TBD	TBD	>30.500	TBD
Special task				
Unmanned combat AV (UCAV)	>1.000	1.500	12.000	2
Lethal (LET)	TBD	300	4.000	3-4
Decoys (DEC)	150-250	0-500	50-5.000	<4

¹Varies with national legal restrictions

Authorities NAAs for certification and airworthiness purposes, is based on the UAV Maximum Take-Off Weight MTOW, and it is equivalent to its counterpart on manned aerial vehicles (Dalamagkidis et al., 2008), (Welch and Bishop, 2005). This classification has been used by NAAs to make the first attempt for categorizing the sector. However there is not a unique global regulation. Moreover, the operation of UAS depends on the different local and governmental agencies involved in the regulation of the sector. Some of the most important NAAs that currently are involved on the regulation of UAS are: The Federal Aviation Administration FAA, that is the agency that regulates the civil aviation on the United States. The Civil Aviation Authority CAA is the equivalent NAA in the United Kingdom and the Civil Aviation Safety Authority CASA is the responsible authority in Australia. The European Aviation Safety Agency EASA is responsible of the type-certification in the European Union in coordination with the NAAs of every member state. Each of this control agencies has its own classification method based on MTOW. Depending on every state, it generally allows a class of UAV with a MTOW below 150 kg, defined as light UAV, that under current regulations, do not require for an airworthiness certification (Dalamagkidis et al., 2008), (European Commission, 2009). UAVs above 150 kg are regulated by National Aviation Authorities NAAs or EASA in the case of the European Union, and should be required to comply with a defined code of airworthiness requirements to achieve the denominated "Type Certification". The code of requirements is chosen according to the kinetic energy, the MTOW and the maximum flight altitude of the aircraft.

In the European Union, for UAVs below 150 kg of MTOW, National Aviation Authorities of the states members have the jurisdiction to certify and control light UAVs. However, there is not enough information about certification requirements for this category. Some regulatory agencies in Europe and UK have proposed some recommendations and guidelines for the operation of light UAVs without obtaining airworthiness certification, subject to the UAV complying with limitations, conditions and procedures similar and at least as demanding as those applied to RC model aircraft (JAA Eurocontrol, 2004), (European Commission, 2009), (Weibel and Hansman, 2009). The UK CAA has also given some recommendations that are consistent with the general operational rules for light UAVs. It is defined in the "Policy for light UAS Systems" (Haddon and Whittaker, 2004). In the same context, the Australian CASR regulates the use of light UAVs with the advisory circular CASR AC 101-1. Therefore, light UAVs covered by these guidelines are those aircraft with a maximum take-off mass below 150 kg, a maximum speed not exceeding 70 kts (129.64 km/h) with a kinetic energy level on impact of less than 95 kJ, operated within 500 meters of a UAV-pilot and not more than 400 ft (121.92 m) above ground level. The 500 m - 400 ft limitation ensures that the human exercising control over the UAV can monitor the airspace and participate in preventing collisions (Sense&Avoid). The 70 kts limit is consistent with staying within 500

m - 400 ft and limiting the impact energy and the capability of a vehicle to harm third parties. Beyond the 500 m - 400 ft boundaries the ability of the human to exercise visual control over a UAS in preventing collisions cannot be relied upon. Small UAVs that go beyond the 500 m - 400 ft limits need to have an effective and reliable Sense&Avoid capability and must be certified.

Currently, the civilian market of UAS above 150 kg is restrained by the absence of standards and regulations, causing that the development and growing of this market will be bounded to the introduction of clear airworthiness certifications and regulations for the integration of UAVs in National Airspace System NAS (European Commission, 2009). As explained by Frost&Sullivan (Frost and Sullivan, 2008), as soon as these regulations are introduced (expected for the middles of the 2010s), there will be an exponential growth of the market as shown in Figure 1.1. However, the lack of a clear regulation or above mentioned guidelines for light UAVs (under 150 kg,) has allowed the design, manufacturing, commercialization and use of different kind of civilian light UAVs during the last years. Currently, there is a growing interest in the use of UAVs in civilian applications that has generated a commercial growing market in which UAVs have demonstrated their utility for some applications like the ones mentioned in (European Commission, 2007):

- Government:
 - Law enforcement (policy, civil security)
 - Border security
 - Coastguard
- Fire and rescue :
 - Forest fires
 - Earthquake rescue
 - Disaster management
- Energy sector
 - Oil and gas pipelines and infrastructure monitoring
 - Power lines and grid monitoring and maintenance
- Agriculture, forestry and fisheries
 - Environmental monitoring
 - Optimizing use of resources
 - Fisheries protection

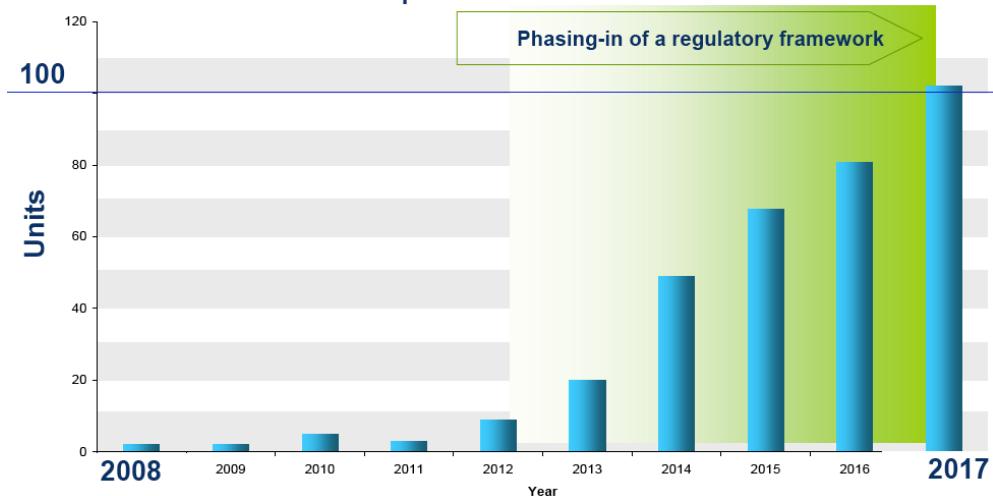


Figure 1.1: The European Civilian UAS Market 2008-2017: Phasing-in of a regulatory framework estimated by Frost&Sullivan.

- Earth observation and remote sensing
 - Climate monitoring
 - Aerial photography, mapping and surveying
 - Major incident and pollution monitoring

According to (European Commission, 2009) and (Frost and Sullivan, 2008) in the short-term, small or mini-UAVs will be the most growing market because they are easier to use within the present air safety regulatory framework. Nevertheless, the development and growth of light UAVs have been marked by low required financial investment, less sophisticated payload requirements and low training burden.

In the market study done by Frost&Sullivan, there is a growing demand in the UAV civil market in future years, specially between 2012 and 2017 (see Figure 1.2). In this market, there is a big prominence of light UAVs, for civilian applications in both governmental and industrial sector. They are required in a variety of applications, like communications, agriculture, fire fighting, energy sector and borders patrol among others.

Although UAS market has demonstrated the potential for a growing sector, there are still considerable efforts to be overcome in order to have UAS operating in non-segregated airspace for a vast range of applications. As mentioned in (European Commission, 2009), in order to have a growing and established market "*UAS must show being more cost effective than current solutions, more*

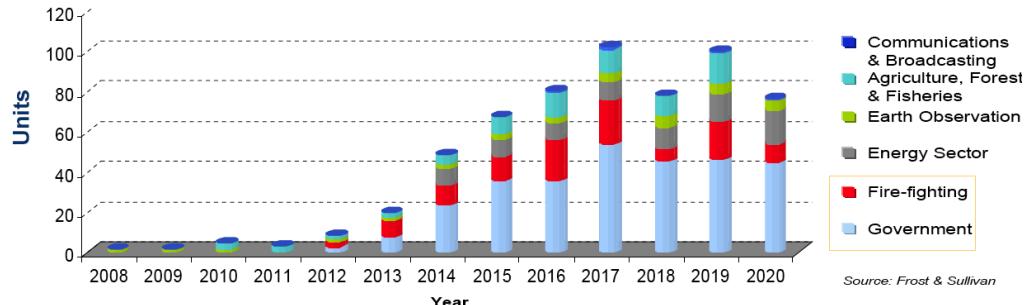


Figure 1.2: The European civilian UAS market 2008-2017 according to the industrial application estimated by Frost&Sullivan.

effective at completing specific tasks, and safe as presently available systems. Finally, they should be able to offer new capabilities that currently do not exist". To accomplish this level of social acceptance and integration in non-segregated airspace, there are several challenges and issues that must be overcome. Some of these issues are related with economic, political and technological factors. The political and economical factors comprise financial support for R&D programs, for the economic stimulation of the industry, regulations and certifications among others, that are beyond the focus of this thesis. However, a comprehensive analysis of the market can be found in (European Commission, 2009) and (Frost and Sullivan, 2008).

In the technological field, there are a number of critical challenges where technologies like Sense&Avoid, secure data links, payloads and systems integration, exploration of novel aerodynamic and propulsion solutions can be highlighted. The development of an airworthy platform is one element that is generally considered as less challenging for manufacturing companies that tended their focus on other aspects of the UAS. Nevertheless, numerous industries, academic research departments and other entities are currently working on R&D projects for these UAS technologies and/or the proposal of novel applications for their use. Such works range from aerodynamics and airframe through payloads and propulsion systems to software and specialized systems and applications design.

As mentioned in (Nonami et al., 2010) currently MALE and HALE UAVs are remotely controlled from a ground control station by a human operator that use basic guidance systems based on following a predefined set of waypoints. These control methods are enough in order to guarantee safe operation of UAVs at high altitudes for reconnaissance and exploration. However light UAVs are generally designed to work at low altitude in cluttered and dynamic environments. Therefore, advanced guidance and navigation capabilities are necessary to detect obstacles, map the environment and plan the flight path.

Sense&Avoid is also one of the vital technologies that is important for UAS integration in non-segregated civilian airspace. This technology, must guarantee the capability of a UAV to prevent mid-air collisions and ensure a safe deconfliction of UAVs that fly with others manned and unmanned aircrafts. It includes the integration with current Air Traffic Management and Control Systems (ATM / ATC) in order to non-saturate the current workload capacity.

Novel applications must also be developed in order to make UAS attractive in a vast range of applications and civilian use. These applications are often restricted to the capability of a UAV to carry light payloads and sensors (cost, size and weight). However, it is common to find that current UAVs have integrated some kind of visual sensor and system, ranging from low cost digital cameras, to sophisticated infrared IR sensors and gimbal systems. These cameras are often used as an "eye in the sky", by transmitting a video to a ground operator. Nevertheless, they can also be used for increasing the capabilities of UAVs, exploiting the vast amount of information that can be supplied by the visual system. The key factors that motivate this thesis are; using visual information, and the integration of UAV control, which is focused on developing novel and practical applications. The key factors are further explained below.

1.2. Problem Statement

The increase in the demand for civil applications of light UAV and constant optimization of computational power and hardware size and weight reduction, have made computer vision a decisive key factor in using light UAV in commercial and governmental civilian applications. In this way, many current UAVs have the capability of carrying professional day light and IR cameras, streaming the video to a ground station operator. However, the large amount of information that can be obtained from a live streaming video is not profited. This information can be used as an accurate, high performance, and multifunctional data suitable for UAV control, navigation, detection, identification, and tracking, in addition to other common features like an eye in the sky or a feedback of an aircraft behavior and performance.

A common usage of visual data provided by a single camera is the analysis of sequential images in order to identify, locate, and continuously follow a moving interest point, region or defined object (multiple elements are also possible) over time on the image. This process is commonly defined as **visual tracking**. There is a large diversity of tracking methods from which approaches can be mention based on features (Mikolajczyk and Smid, 2005), direct methods (correlation) (Irani and Anandan, 2000), color and shape algorithms among others. The goal of any visual tracking system is to be able to identify a reference pattern correctly and continuously on the image plane independently from the variations

presented on the image sequence with respect to parameters like camera or scene rotations and translations, object occlusions, illumination changes and noise, among others. Visual tracking is the basis for the development of more complex methods, like the ones that use imagery analysis for position and orientation estimation of a robot, in a process referred to as **visual odometry**. In these methods, sequential images are analyzed in order to extrapolate the robot space movement based on the image motion. The obtained motion, can be used as a main odometer sensor or fused with information from other sources such as GPS, inertial sensors, wheel encoders in order to have a refined robot motion estimation (Corke et al., 2004),(Nistér et al., 2006).

Image processing is also employed in **visual navigation** tasks that can be defined as the use of visual data in the process of determining objects position (robot and scene) and a suitable and safe path between a starting and a goal point for a robot traveling between them (Bonin-Font et al., 2008). Depending on the scene knowledge, visual navigation can be classified as map-based or map-building systems. In map-based methods robot make a self-localization process, in which features in sequential images are matched with a known map (landmarks) in order to estimate and update the robot position. Meanwhile, in map-building systems, the robot first creates a visual features scene map when it navigates, and then uses it for self-localization using feature matching. If this process is done simultaneously, it is known as visual Simultaneous Localization and Mapping or **visual SLAM** (Montiel et al., 2006). It is often based on algorithms that creates an “on-air” probabilistic feature-based map that evolves continuously and dynamically taking into account camera motion and the observation of actual and new features (Davison et al., 2007).

Finally, **visual servoing** also known as vision-based robot control is a technique which uses feedback information extracted from a vision sensor to control the motion of a robot (Hutchinson et al., 1996), (Chaumette and Hutchinson, 2006). In this context, visual data is used to estimate either the camera position in the workspace or the features on the image plane, using it for defining an error control function that is related with the camera-robot dynamics, in order to command robot actuators to accomplish a goal.

Image processing, is a hardware and software demanding activity that is generally performed by specialized hardware, not suitable to be implemented in a light UAV. However, in these last years, there has been a considerable development in the use of computational resources and the miniaturization of the hardware that have permitted the implementation of simple vision algorithms on-board mini UAVs. In addition, a vision processing algorithm to be useful for a UAV, must fulfill a series of requirements such as real time and precision performance. Currently, UAV control systems run at frequencies above 100 Hz, making mandatory, that any image processing algorithm used as a sensor must

be capable of running at frequencies comparable to the one controller demands information.

The necessity for implementing computer vision algorithms suitable to running on a limited performance hardware on-board a light UAV and capable of giving accurate information for a control unit, defines the research framework from which this thesis has been developed. This research framework, wants to develop and demonstrate the usefulness of computer vision as an accurate and fast sensor for a series of functions like attitude and position estimation, obstacle detection and evasion, object tracking and identification, among others. In addition, this information is employed to increase the autonomous control capabilities of a light UAV in difficult and demanding tasks such as landing, accurate positioning and tracking.

1.3. Thesis proposal and objectives

The main goal of the computer vision group (CVG-UPM, 2010) in the field of aerial robotics, is the use of Unmanned Aerial Vehicles guided and controlled by computer visual systems for applications in non-segregated civilian spaces. This thesis aims to improve the state of the art in this growing field by making contributions to this general proposal considering the following objectives.

- To estimate the flight attitude (*Roll, Pitch, Yaw*), 3D pose and motion using only visual information and computer vision techniques, as a principal sensor or as redundant system suitable for use on control systems for UAV.
- To implement robust computer vision techniques for visual identification, matching and tracking capable for use on applications and tasks like visual tracking and servoing, state estimation, pose and flight control.
- To apply visual processing techniques for UAV flight control tasks like autonomous hovering, landing on visual landmarks, static and mobile objects identification, tracking and servoing.
- To generate visual control methods that can be used on diverse systems and a wide range of UAVs and on-board visual systems.

1.4. Thesis Contributions

This thesis presents an improvement on the state of the art in visual control application for UAVs. The thesis exploits the large amount of information provided by on-board visual system generating a framework in which is

demonstrated how to estimate the aircraft attitude, position, and motion only based on visual information. Moreover, it also shows how this state information is used for making precision maneuvers in order to control a UAV in a variety of tasks like positioning, tracking, hovering and landing. In addition, a series of general proposed visual algorithms like mosaic building and video stabilization are presented and improved for micro and small UAVs.

The main contributions of this thesis are summarized below:

■ Omnidirectional visual systems in UAV applications:

The unified theory for central catadioptric cameras to model large FOV dioptric and catadioptric cameras, has been employed to increase the FOV of the aircraft without increasing the complexity of a visual system. Benefits of these systems have been demonstrated by means of two applications that have been experimentally validated:

- The use of a **calibrated catadioptric camera for an on-board attitude and heading estimation**, showing how the skyline projection on the catadioptric image is segmented and used to calculate attitude, and how the appearance images obtained from the omnidirectional images are used to obtain a visual compass, that is a useful tool to calculate the relative rotation and heading of the aerial vehicle.
- The use of a **calibrated fisheye lens in order to define a See&Avoid architecture** is demonstrated, showing how the proposed contribution based on object tracking and a visual servoing scheme defined on the unitary sphere can be used to command a UAV in order to make an efficient and safe collision avoidance maneuver in conflictive flying situations.

■ Tridimensional pose estimation based on visual tracking:

A contribution for UAV 3D pose estimation using planar object tracking is proposed and validated. The method exploits the rich information obtained by a projective transformation of planar objects on a calibrated camera. The algorithm obtains the metric and projective components of a reference object (landmark or helipad) with respect to the UAV camera coordinate system (allowing obtaining the relative position and attitude of the aircraft with respect to a landmark), using a robust object tracking based on homographies and feature tracking.

■ Visual control using decoupled architectures: Different methods in order to develop high level decoupled visual servoing architectures that directly relates controlled variables with unique input features are presented and validated. The proposed methods have an excellent robustness to drive the non-holonomic behavior of rotary wing vehicles, as well as the perturbations commonly presented in outdoors flights as demonstrated by the following control schemes:

- **UAV control based on visual 3D pose estimation:**

Proposal and validation of a aircraft control algorithm based on the position and attitude of the aircraft with respect a landmark in order to make accurate positioning, hovering, and landing, closing a high level control loop using a *dynamic position base look-and-move* architecture.

- **UAV control based on visual features:**

Classic approaches for visual servoing are extended to cover diverse kinds of cameras (pinhole and omnidirectional) generating control structures that allows to make practical control systems like moving objects following and See&Avoid systems for light UAV.

- **UAV applications based on visual information and image processing:**

The general projective model for perspective images and the unified theory for omnidirectional images are exploited to make general algorithms for a UAV, like mosaic building, video stabilization, panoramas and spherical image construction, visual tracking among others. The techniques have been implemented in order to be used on-board micro and small UAVs with different series of visual systems, like conventional perspective cameras, eye fish lenses, and catadioptric systems.

1.5. Dissertation outline

Chapter 2 presents an introduction of the status quo of computer vision processing algorithms used in UAVs applications.

Chapter 3 presents a review of image processing algorithms used for visual matching and tracking. It also explains how these algorithms are employed for robust projective transformations estimation, that are then used for obtaining the camera attitude and position in the workspace.

Chapter 4 presents applications of the projective models and the visual tracking on common duties on UAVs, including applications for video stabilization, mosaicing building, and UAV position estimation.

In Chapter 5, different vision processing algorithms for tracking, pose and attitude estimation are integrated on-board aircraft in order to close the control loop of a UAV using different system architectures for tasks such as positioning, landing, and object following.

Chapter 6 presents the unified model for central catadioptric cameras and the use of omnidirectional images on-board a UAV. The projection model of catadioptric images on the unitary sphere is utilized for generating a visual compass and estimating aircraft attitude based on the segmentation of the

skyline. It is also employed for a novel See&Avoid application for light UAVs using dioptric systems modeled with the unified projection model and visual servoing techniques on the unitary sphere.

Finally chapter 7 presents conclusions and future work.

Chapter 2

State of the Art

2.1. Introduction

Unmanned Aerial Vehicles is an active research area in many universities and R&D institutions around the world. They have been used as testbeds for investigating a vast variety of problems and applications, ranging from motion, pose and attitude estimation, control, navigation and path planning to object detection and tracking, as well as navigation. In many of them, computer vision has played a predominant function as fast, reliable and accurate source of information, in addition of being an “eye in the sky” suitable for a vast range of applications such as imagery capturing, inspection, vigilance, identification, tracking among others (Campoy et al., 2009), (Nikolos et al., 2004), (Caballero et al., 2009), (Golightly and Jones, 2005). Section 2.2 presents a brief review of the state of the art of computer vision applied on UAVs.

The UAV history, development, architectures and flight control theory are out of the focus of this thesis. The *Aerial Robotics* chapter (Handbook of robotics) (Siciliano and Khatib, 2008) gives a general review of the UAVs history, flight concepts and applications. Readers are also referred to (Valavanis and Kontitsis, 2007), in which a historical perspective review of UAVs is presented.

Around the world, there are research institutions, principally from the

academic field that have been working actively on the research and development of vision systems for UAVs. Research in the field of visual control for aerial vehicles demand a large R&D infrastructure in terms of personal resources, infrastructure, hardware and specialized equipment, that are often only possible in large research groups like the ones belonging to universities or national and public research institutions. Meanwhile, private industrial sectors have also focused on UAS's. However, because of the commercial and private business, there is not much public information about the state of the art. Appendix C presents a summary of some of the most outstanding groups around the world.

2.2. Computer Vision for UAV

Computer vision has demonstrated being a powerful, non intrusive and low-cost sensor useful for many applications in robotics and control system. It has a large number of applications and constantly, there are new developments that reduce cost, size, weight, but increasing the capacity for acquiring and processing high resolution images with an adequate frequency to be used for control system on mobile robotics.

There is not a unique classification for the use of visual processing on UAVs. On (Ollero and Merino, 2004), they present some of the methods and technologies that have been applied in aerial robotics, making a brief review of visual applications on UAVs. Liu (Liu and Dai, 2010) makes a survey of computer vision techniques applied to UAV flight control and perception, focusing on visual navigation, aerial surveillance and visual SLAM.

Perhaps the most common way to classify the use of computer vision on UAVs, is based according to the degree of interaction with the aerial aircraft, ranging from a simple applications like image capturing and monitoring (**surveillance**) passing throughout its use to pose estimation and navigation (**visual odometry**) up to its integration in the aircraft control loop (**visual servoing**) for applications like landing, positioning, tracking, among others.

Aerial photography acquisition could be the most basic application of computer vision on UAVs (Cui et al., 2008). It is currently the most active commercial use of UAVs, because it is a low cost, fast and reliable system for applications like high resolution photography (Myers and Miller, 2005), or traffic monitoring and data collecting as has been showing by Valavanis (Puri et al., 2007). Visual systems on aerial vehicles have also demonstrated being a fast system for acquiring high resolution imagery during emergency situations (Lewis, 2007) including natural disasters (Quaritsch et al., 2010), fire detection (Merino et al., 2006), (Nikolos et al., 2004) among others.

A natural extension of aerial imagery is its use in **aerial photogrammetry** (Eisenbeiss, 2004) in which geo-referenced (GPS+IMU) images are used to obtain some metric of objects in the scene. Applications like surface mapping, ortographics maps, aerial image 3D viewing among others can be developed. A good review of UAV photogrammetry, applications and examples in archeological research is presented in (Eisenbeiss, 2009).

UAV photogrammetry can also be used to refine the pose estimation of a UAV based on satellite images. Cesetti (Cesetti et al., 2011) presents a method for aerial vehicles pose estimation based on the registration of images captured by a UAV and a set of geo-referenced satellite images.

Visual processing can also be used for estimating aircraft space movements, based on the changes presented on the acquired images, allowing the detection and measurement of small position changes that are not measurable with other sensor because their spatial resolution(i.e GPS accuracy). The process of motion estimation from imagery is often referred as **visual odometry** (Matthies, 1989). It determines a vehicle's position and orientation by detecting and tracking salient points using an on-board camera. This technique has been traditionally used by ground vehicles to improve robot localization. The use of visual odometry techniques has been documented in different situations such as, Mars rovers (Corke et al., 2004), and (Cheng et al., 2006). Image-based motion estimation to augment inertial navigation for safe and precise landing on planetary bodies is described in (Roumeliotis et al., 2002).

In the case of aerial vehicles, one of the first stereo systems used for an autonomous helicopter is described in (Amidi, 1996). This system uses a customized vision processing hardware to estimate a helicopter's position. A comparison of two altitude estimation approaches using stereo and sonar is given in (Corke et al., 2000). In the work presented by Johnson (Johnson and Matthies, 1999), a technique for attitude and position estimation that will enable the precision guidance necessary for autonomous small body landing using descent camera imagery and laser altimetry is described. Bosse (Bosse, 1997) uses an extended Kalman filter to merge IMU, GPS, sonar altimeter and camera motion estimates into a navigation system for an autonomous helicopter. Mejias (Mejias et al., 2007) uses a stereo system to measure the motion and height of a rotary wing UAV flying at low altitudes. Recently, a visual odometer has been implemented in a small rotorcraft UAV (Kendoul et al., 2009), by identifying and tracking visual features in the environment, obtaining optical flow that is then fused with the inertial measurements in order to determine the position and velocity of the aircraft.

Visual information has also been used for **UAV attitude estimation**. The first experiments were based on images taken by an aircraft with a forward

looking monocular camera. These images were used to estimate the *Roll* angle with an horizontal reference based on the skyline segmentation (Ettinger et al., 2002), (Todorovic et al., 2003), (Cornall et al., 2006), (Cornall and Egan, 2004), (Dusha et al., 2007). The work presented in (Todorovic et al., 2003) deal with the horizon detection problem as image segmentation and object recognition, applying statistical appearance models based on both color and texture clues. They make a Bayesian segmentation based on a statistical framework employing Hidden Markov Trees on the appearance models. Cornall (Cornall et al., 2006) uses a technique focused on be implemented on a small micro controller with low computation power. The algorithm uses a method where the horizon is segmented by thresholding the blue color plane (RGB color space), determining the optimal threshold by Otsu's method (Otsu, 1979) and Kmeans. This approximation gives goods results under clear sky conditions. Dusha (Dusha et al., 2007), proposed a method for UAV *Roll* and *Pitch* estimation based on the segmentation of the horizon line using a Hough transformation, that also uses the optical flow of the skyline in order to estimate the body frame rates. Finally, in (Thurrowgood et al., 2009) a method for attitude estimation have been presented in which the horizon identification is based on color and intensity segmentation.

Simultaneous Localization and Mapping based only on visual information (**visual SLAM**) have also been implemented on UAVs. Adrien *et. al.* (Adrien et al., 2006) have presented a method for purely vision-based 2D SLAM for micro UAV flying at a constant altitude. The system is capable of building a visual map based on feature matching and correctly locate a UAV in this map. Artieda *et. al.* (Artieda et al., 2009) showed the applications of monocular visual SLAM on rotary wing UAV by building a 3D maps of objects in the image w.r.t. a UAV. Caballero (Caballero et al., 2009) presents a similar approach based on mosaics. Nemra (Nemra and Aouf, 2009) proposed a visual SLAM approach based on a stereo vision system.

The information derived from the use of a visual sensor on-board UAV can be used to feedback an aircraft autopilot, in order to increase the capabilities, robustness and reliableness of a control system. Visual information in the control loop commonly referenced as **visual servoing** (Hutchinson et al., 1996), (Chaumette and Hutchinson, 2006), (Chaumette and Hutchinson, 2007), has the main advantage of providing refined information about UAV position and attitude as well as a dynamic and timely data like the presence of objects within the UAV field of view and its relative position w.r.t the aircraft. This information can successfully complement and in some cases replace other classical sensors for UAV navigation such as GPS and IMU, which provide a very different type of information, i.e. absolute coordinates that are not related to the position of nearby objects (Achtelik et al., 2011), (Salazar et al., 2009).

Visual servoing can be used in order to control either a pan and tilt camera platform (servo controlled or gyro stabilized) or directly the aircraft (a combination of both is also possible). Pan and tilt camera platforms gives more freedom of movements to the UAV, reducing the aircraft movements needed to track objects. There are some tests using a pan and tilt camera-platform, like the one present by Zou (Zou et al., 2006), in which a method for UAV camera pan and tilt inspired on the human eye is used on UAV for ground moving target tracking. In the same line, Dobrokhodov (Dobrokhodov et al., 2006), presented a method for moving object tracking using a pan and tilt camera platform on a fixed wing UAV that also uses visual information for estimating the GPS world coordinates of the moving target, as well as the visual servoing method for UAV camera pan and tilt platform presented by Lin (Lin et al., 2009)

In a different way, aircraft visual servoing is employed to directly maneuver an aerial vehicle in order to perform a large variety of tasks such as aircraft leveling and stabilization, hovering and positioning, landing and takeoff, object following and navigation, collision avoidance, refueling among others. Ettinger (Ettinger et al., 2002) proposed a method for fixed wing UAV stabilization control (*Roll*) based on the detection of the skyline. Different works have also been done where a vision system was used for low altitude position estimation and autonomous landing. In the work presented in (De Wagter and Mulder, 2005), the authors have evaluated the use of visual information at different stages of a UAV control system, including a visual controller and a pose estimation for autonomous landing using a chessboard pattern. Merz (Merz et al., 2004), (Merz et al., 2006), uses a method that fuses visual and inertial information in order to control an autonomous VTOL aircraft landing on known landmarks. Saripalli *et. al.* have proposed an experimental method for autonomous landing on a moving target, (Saripalli and Sukhatme, 2007), (Saripalli et al., 2003), by tracking a known helipad and using it to complement the controller IMU+GPS state estimation. In (Fucen et al., 2009) a visual system is used to detect, identify a landing zone (helipad) and confirm the landing direction of the vehicle, have been presented.

Vision based landing for multi-rotor UAVs has been an actively studied field in recent years. Some examples are the work presented by Lange in (Lange et al., 2008) where the visual system is used to estimate a vehicle position relative to a landing place. Recently, Nonami (Nonami et al., 2010) and then Wenzel (Wenzel et al., 2011) have presented two different methods for small UAV autonomous takeoff, tracking and landing on a moving platform. The first is based on optical flow, the second uses IR landmarks visual tracking to estimate the aircraft position.

Visual servoing has also been used for accurate positioning and hovering with applications to inspection and surveillance. Mejias (Mejias et al., 2006)

presented a method based on feature tracking in order to control the lateral and vertical displacement of a rotary wing UAV in urban scenarios. In (Metni and Hamel, 2007) an application for bridge inspection based on visual servoing of a planar landmark and the decomposition of a homography for pose estimation is presented. In (Salazar et al., 2009) a method for small multi-rotor UAV control based on a stereo visual servoing is presented. The method used a stereo system to detect and track a landmark and to estimate the 3D position of the UAV. This information is then used to make a vision based hovering control.

Visual information can also be used for aircraft guidance, path planning and trajectory following. Hrabar (Hrabar et al., 2005) proposed a system based on optical flow on stereo images for maneuvering close to fixed objects in urban scenarios (Hrabar et al., 2005). In (Rathinam et al., 2006) presents an autonomous system for structures (highway, bridges, pipelines) identification and tracking useful for inspection. Similarly, Egbert (Egbert and Beard, 2010) proposed a method for fixed wing micro UAV for tracking and following a road with a low altitude flight. The method uses computer vision as the primary guidance sensor, detecting the road using connected components and color segmentation on hue-saturation-value HSV color space. Ander (Andert et al., 2010) presented a method that uses visual information as an aid for navigation of a UAV through gates with known global position. Visual system is used for detecting the gates and refining the global position in order generate accurate paths throughout the gates.

Applications for UAV guidance and control in emergency situations have also been successfully demonstrated. Carnie (Carnie et al., 2006) showed that visual systems on-board aerial vehicles can be used to detect airborne traffic and respond with appropriate avoidance maneuvers in order to maintain minimum safety separation distances. In (De Wagter and Mulder, 2005) the advantages of using visual sensors on-board UAVs in order to increase the capabilities of UAVs as well as using the vision as a redundant system for other classical sensors like GPS and IMU have been demonstrated. In (He et al., 2006) motion field on UAV images is employed to compute a range map of objects in the camera FOV, that is then used for navigation and obstacles avoidance. A visual system inspired by flying insects for detecting and avoiding static structures has been proposed by Beyeler (Beyeler et al., 2009). Lai (Lai et al., 2011) presented a real-time system for Sense&Avoid by using parallel image processing on commercial graphics processing units (GPUs).

Eng (Eng et al., 2010) has demonstrated a visual system for safe landing places identification and guidance under emergency force landings. In the same way, in (Cesetti et al., 2010) a vision-based landing and navigation algorithm for an autonomous helicopter is used for selecting a safe landing area based on satellite images and on-board images using image registration methods based on

visual keypoints matching.

Image processing in large field of view FOV sensors has also been employed in UAVs for applications like control, attitude estimation or collision avoidance, among others. Hrabar (Hrabar and Sukhatme, 2003) uses an omnidirectional system for sideways looking sensing on an autonomous helicopter, generating control commands for a visual servoing task, using the centroid of known visual targets. Conroy (Conroy et al., 2009) uses spatial decompositions of the instantaneous optic flow to extract local proximity information from catadioptric images obtained on-board a micro-air-vehicle MAV for corridor-like environments navigation. Demonceaux (Demonceaux et al., 2006) demonstrated the advantages of using omnidirectional rather than perspective images for attitude estimation. In (Dunbabin, 2007) a catadioptric system designed to use optical flow for measurement and control of height above the ground has been tested.

Based on the state of art, a possible classification of computer vision on UAVs is based on how the image is processed and integrated with other sensors and used on an aircraft. It is composed of three levels, ranging from only image processing up to direct interaction with the UAV. First, common image processing techniques are employed to process raw images captured on-board. It includes techniques like image segmentation, object identification and tracking, camera motion estimation and map building among others. Then, image processing techniques are combined and fused with other sensors on-board aircraft in order to generate defined applications or services. Applications range from simple aerial image collection (geo-referenced) up to aircraft visual navigation. Finally, visual information interacts with the aircraft in order to make specific control actions on a UAV. Figure 2.1 shows a generic scheme for the use of computer vision techniques on UAVs according to this proposed classification.

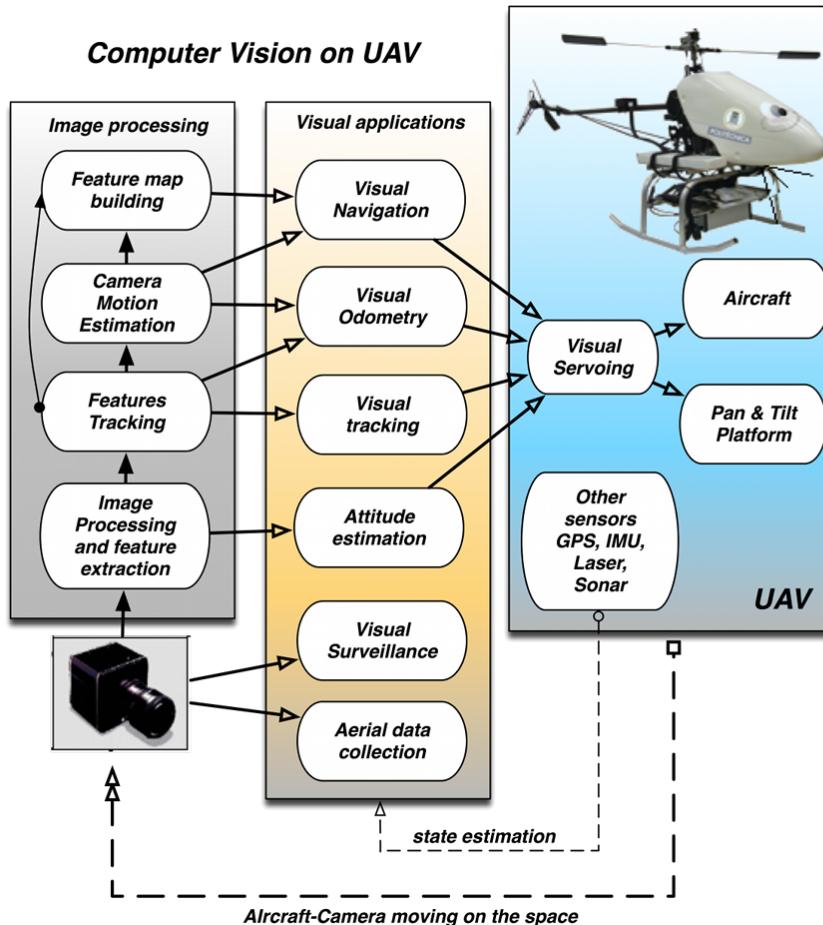


Figure 2.1: Vision on UAVs generic scheme: In the first level, image processing techniques are employed to process raw images captured on-board. In the second level, visual data can be fused with other sensors on-board to make specific applications. Finally, the visual information is used to control a UAV.

Chapter 3

Image Processing for UAV applications

3.1. Introduction

This chapter summarizes most of the image processing techniques employed in this thesis for a variety of applications including a visual control system for a UAV. First, different methods are introduced in order to find relevant visual features and matching methods useful for identifying and tracking regions of interest in image sequences, then, this tracking method is used to estimate a projection model in the camera image space and finally, how it is employed in order to determine the 3D pose and camera movement on the space are shown.

3.2. Image Based Visual Tracking

Image processing is used for finding characteristics in the image that can be applied to recognizing an object or points of interest. This relevant information extracted from the image (called features) ranging from simple structures, such as points or edges, to more complex structures, such as objects. Such features can be used as references for many visual servoing task and control system.

In image regions, the spatial intensity can also be considered as a useful characteristic for patch tracking. In this context, the region intensities are considered as a unique feature that can be compared using correlation metrics in image intensity patterns.

Most of the features used as reference are interest points, which are points in an image that have a well-defined position. They can be robustly detected, and are usually found in any kind of images. Some of these points are corners formed by the intersection of two edges, and others are points in an image that have rich information based on the intensity of the pixels. A detector used for this purpose is the Harris corner detector (Harris and Stephens, 1988). It extracts corners very quickly based on the magnitude of the eigenvalues of the autocorrelation matrix. It is where the local autocorrelation function measures the local changes of a point with patches shifted by a small amount in different directions. However, taking into account that the features are going to be tracked along an image sequence, using only this measure to guarantee the robustness of the corner is not enough. This means that good features to track (Shi and Tomasi, 1994) have to be selected in order to ensure the stability of the tracking process. The robustness of a corner extracted with the Harris detector can be measured by changing the size of the detection window, which is increased to test the stability of the position of the extracted corners. A measure of this variation is then calculated based on a maximum difference criteria. Besides, the magnitude of the eigenvalues is only used in order to keep features with eigenvalues higher than a minimum value. Combination of such criteria leads to the selection of the good features to track. Figure 3.1(a) shows an example of good features to track on an image obtained using a UAV.

The problem of tracking features can be solved with different approaches. The most popular algorithm to track features and image regions, is the Lucas Kanade algorithm (Lucas and Kanade, 1981) which has demonstrated a good speed performance with a good stability for small changes. Recently, feature descriptors have been successfully applied to visual tracking, showing a good robustness for image scaling, rotation, translation, and illumination changes, even though they are time computing expensive to calculate. The generalized Lucas Kanade algorithm is presented in subsection 3.2.1, where it is applied for patch tracking and also for optical flow calculation, using the sparse LKT and pyramidal LKT variations. Subsection 3.2.2 features descriptors are introduced and used for robust matching, as explained in subsection 3.2.3.

3.2.1. Appearance tracking

Appearance-based tracking techniques use the intensity values of a ‘patch’ of pixels that correspond to the object to be tracked. One of these methods to track

such patch of pixels is the generalized LKT algorithm that works under three premises: first, the constancy of intensity: the vicinity of each pixel considered as a feature does not change as it is tracked from frame to frame; second, the change in the position of the features between two consecutive frames must be minimum, so that the features are close enough to each other; and third, the neighboring points move in a solidarity form, maintaining spatial coherence.

A patch is related to the next frame by a warping function that can be the optical flow or another model of motion. Taking into account the previously mentioned LKT premisses, the problem can be formulated in such manner similar to the one presented in (Baker and Matthews, 2002): lets define \mathbf{x} as the set of points that form a patch window or template image $T(\mathbf{x})$, where $\mathbf{x} = (x, y)^T$ is a column vector with the coordinates in the image plane of a given pixel and $T(\mathbf{x}) = T(x, y)$ is the grayscale value of an images at alocations \mathbf{x} . The goal of the algorithm is to align the template T with the input image I (where $I(\mathbf{x}) = I(x, y)$ is the grayscale value of the images at a location \mathbf{x}). Because T transformed must match with a sub-image of I , the algorithm will find a set of parameters $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ for a motion model function (e.g., optical flow, affine, homography) $W(\mathbf{x}; \mu)$, also called the warping function. The objective function of the algorithm to be minimized in order to align the template and the actual image is presented in Equation 3.1:

$$e(\mathbf{W}) = \sum_{\forall \mathbf{x} \in X} [I(W(\mathbf{x}; \mu)) - T(\mathbf{x})]^2 w(\mathbf{x}) \quad (3.1)$$

where $w(\mathbf{x})$ is a function to assign different weights to the comparison window. In general $w(\mathbf{x}) = 1$. Alternatively, $w(\mathbf{x})$ could be a gaussian function to emphasize the central area of the window. This equation can also be reformulated to make it possible to solve for tracking sparse features as is explained in Section 3.2.1.

Lucas Kanade problem is formulated to be solved in relation to all features in the form of a least squares problem, having a closed form solution as follows.

Defining $w(\mathbf{x}) = 1$, the objective function (Equation 3.1) is minimized with respect to μ and the sum is performed over all of the pixels \mathbf{x} on the template image. Since the minimization process has to be made with respect to μ , and there is no lineal relation between the pixel position and its intensity value, Lucas Kanade algorithm assumes a known initial value for the parameters μ and finds increments of the parameters $\delta\mu$. Hence, the expression to be minimized is:

$$\sum_{\forall \mathbf{x} \in X} [I(W(\mathbf{x}; \mu + \delta\mu)) - T(\mathbf{x})]^2 \quad (3.2)$$

and the parameter update in every iteration is $\mu = \mu + \delta\mu$. In order to solve Equation 3.2 efficiently, the objective function is linearized using a Taylor Series

expansion employing only the first order terms. The parameter to be minimized is $\delta\mu$. Afterwards, the function to be minimized looks like Equation 3.3 and can be solved like a least squares problem with Equation 3.4.

$$\sum_{\forall \mathbf{x} \in X} \left[I(W(\mathbf{x}; \mu)) + \nabla I \frac{\partial W}{\partial \mu} \delta\mu - T(\mathbf{x}) \right]^2 \quad (3.3)$$

$$\delta\mu = \mathbf{H}^{-1} \sum_{\forall \mathbf{x} \in X} (\nabla I \frac{\partial W}{\partial \mu})^T (T(\mathbf{x}) - I(W(\mathbf{x}; \mu))) \quad (3.4)$$

where $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ is the gradient of image I evaluated at $W(\mathbf{x}; \mu)$, the term $\frac{\partial W}{\partial \mu}$ is the Jacobian of the warp, and \mathbf{H} is the Hessian Matrix approximation, defined as Equation 3.5 shows:

$$\mathbf{H} = \sum_{\forall \mathbf{x} \in X} (\nabla I \frac{\partial W}{\partial \mu})^T (\nabla I \frac{\partial W}{\partial \mu}) \quad (3.5)$$

More details about this formulation can be found in (Buenaposada et al., 2003) and (Baker and Matthews, 2002), where some modifications are introduced in order to make the minimization process more efficient, by inverting the roles of the template and changing the parameter update rule from an additive form to a compositional function. This is the so called Inverse Compositional Image Alignment ICIA algorithm, first proposed in (Baker and Matthews, 2002). These modifications were introduced to avoid the cost of computing the gradient of the images, the Jacobian of the warping function in every step and the inversion of the Hessian matrix that involves the most computational cost of the algorithm.

Sparse Lucas Kanade

The Lucas Kanade algorithm can be applied on small windows around distinctive points as a sparse technique. In this case, the template is a small window (i.e., size of 3, 5, 7 or 9 pixels) and the warping function is defined by only a pure translational vector. In this context, the first assumption of the Lucas Kanade method can be expressed as follows: given a point $\mathbf{x}_i = (x, y)$ in the image plane at time t which intensity is $I(x, y, t)$ will have moved by v_x , v_y and Δt between the two image frames, the following equation can be formulated:

$$I(x, y, t) = I(x + v_x, y + v_y, t + \Delta t) \quad (3.6)$$

If the general movement can be consider small and using the Taylor series expansion, Equation 3.6 can be developed as:

$$I(x + v_x, y + v_y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} \Delta t + HOT. \quad (3.7)$$

Because the higher order terms HOTS can being ignored, Equation 3.7 can be expressed as:

$$\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t}\Delta t = 0 \quad (3.8)$$

where v_x, v_y are the x and y components of the velocity or optical flow of $I(x, y, t)$ and $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ and $I_t = \frac{\partial I}{\partial t}$ are the derivatives of the image at point $\mathbf{x} = (x, y, t)$

$$I_x v_x + I_y v_y = -I_t \quad (3.9)$$

Equation 3.9 is known as the *aperture problem* of the optical flow. It arises when you have a small aperture or window to measure motion. If motion is detected in this small aperture, it is often that it will be seeing as an edge and not as a corner, and thereby causes that movement direction not to be determined. In order to find the optical flow, another set of equations is needed, given by some additional constraint.

Lucas Kanade algorithm forms an additional set of equation assuming that there is a local small window of size $m \times m$ centered at point $p = (x, y)$ in which all pixels moves coherently. If the pixels within a windows are enumerated as $1, 2, \dots, n$, with $n = m^2$, a set of equations can be formulated as:

$$\begin{aligned} I_{x_1} v_x + I_{y_1} v_y &= -I_{t_1} \\ I_{x_2} v_x + I_{y_2} v_y &= -I_{t_2} \\ &\vdots \\ I_{x_n} v_x + I_{y_n} v_y &= -I_{t_n} \end{aligned} \quad (3.10)$$

Equation 3.10 has more than two equations for the two unknowns and thus the system is over-determined. A system of the form $\mathbf{Ax} = \mathbf{b}$ can be formed as Equation 3.11 shows.

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -I_{t_1} \\ -I_{t_2} \\ \vdots \\ -I_{t_n} \end{bmatrix} \\ \text{or} \\ \begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} &= \begin{bmatrix} -I_{t_1} \\ -I_{t_2} \\ \vdots \\ -I_{t_n} \end{bmatrix} \end{aligned} \quad (3.11)$$

The least squares method can be used to solve the over-determined system of Equation 3.12, finding that the optical flow can be defined as:

$$\begin{aligned} \mathbf{A}^T \mathbf{A} \mathbf{x} &= \mathbf{A}^T \mathbf{b} \\ \text{or} \\ \begin{bmatrix} v_x \\ v_y \end{bmatrix} &= (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \end{aligned} \tag{3.12}$$

Pyramidal Lucas Kanade

In images with high motion, good matched features can be obtained using the Pyramidal Lucas Kanade algorithm modification (Bouguet Jean Yves, 1999). It is used to solve the problem that arises when large motion is presented between consecutive frames. It is done by first tracking features over large spatial scales on the pyramid image, obtaining an initial motion estimation, and then refining it by down sampling the levels of the images in the pyramid until it arrives to the original scale.

The overall pyramidal tracking algorithm proceeds as follows: first, a pyramidal representation of an image I of size $width \times height$ pixels is generated. The zeroth level is composed by the original image and defined as I^0 , then pyramids levels are recursively computed by downsampling the last available level (compute I^1 from I^0 , then I^2 from I^1 and so on until I^{L_m} from I^{L-1}). Typical maximum pyramid levels L_m are 2, 3 and 4. Then, the optical flow is computed at the deepest pyramid level L_m . Then, the result of that computation is propagated to the upper level $L_m - 1$ in a form of an initial guess for the pixel displacement (at level $L_m - 1$). Given that initial guess, the refined optical flow is computed at level $L_m - 1$, and the result is propagated to level $L_m - 2$ and so on up to the level 0 (the original image).

3.2.2. Feature descriptors and tracking

Feature description is a process of obtaining interest points in the image which are defined by a series of characteristics that make it suitable for being matched in image sequences. These characteristics can include a clear mathematical definition, a well-defined position in image space and a local image structure around the interest point. This structure has to be rich in terms of local information contents that have to be robust under local and global perturbations in the image domain. This robustness includes those deformations arising from perspective transformations (i.e., scale changes, rotations and translations) as well as illumination/brightness variations, such that the interest points can be reliably computed with high degree of reproducibility.

There are many feature descriptors suitable for visual matching and tracking, from which Scale Invariant Feature Transform SIFT and Speeded Up Robust Feature algorithm SURF are some of the most widely used. They are presented in following two sections.

SIFT features

The Scale Invariant Feature Transform detector (Lowe, 2004) is one of the most widely used algorithms for interest point detection (called keypoints in the SIFT framework) and matching. This detector was developed with the intention to be used for object recognition. Because of this, it extracts keypoints invariant to scale and rotation using the gaussian difference of the images in different scales to ensure invariance to scale. In order to achieve invariance to rotation, one or more orientations based on local image gradient directions are assigned to each keypoint. The result of all this process is a descriptor associated to a keypoint, which provides an efficient tool to represent an interest point, allowing an easy matching against a database of keypoints. The calculation of these features has a considerable computational cost, which can be assumed because of the robustness of the keypoint and the accuracy obtained when matching these features. However, the use of these features depends on the nature of the task: whether it needs to be done fast or accurate. Figure 3.1(b) shows an example of SIFT keypoints on an aerial image taken with a UAV.

SIFT features can be used to track objects. The rich information given by the keypoint descriptors are the ones considered. An object is matched along the image sequence comparing the model template (the image from which the database of features is created) and the SIFT descriptors of the current image, using the nearest neighbor method. Given the high dimensionality of the keypoint descriptor (128), its matching performance is improved using the Kd-tree search algorithm with the Best Bin First search modification proposed by Lowe (Beis and Lowe, 1997). The advantage of this method lies in the robustness of the matching using the descriptor and also in the fact that this match does not depend on the relative position of the template and the current image. Once the matching is performed, a perspective transformation is calculated using the matched keypoints, comparing the original template with the current image.

SURF Features

Speeded Up Robust Feature algorithm (Herbert Bay et al., 2006) extracts features from an image which can be tracked over multiple views. The algorithm also generates a descriptor for each feature that can be used to identify it. SURF feature descriptor is scale and rotation invariant. Scale invariance is attained using different amplitude gaussian filters, in such a way that its application

results to an image pyramid. The level of the stack from which the feature is extracted assigns the feature to a scale. This relation provides scale invariance. The next step is to assign a repeatable orientation to the feature. The angle is calculated through the horizontal and vertical Haar wavelet responses in a circular domain around the feature. The angle calculated in this way provides a repeatable orientation to the feature. This relationship is used to attain the descriptor's angle invariance. Figure 3.1(c) shows an example of SURF features in an aerial image.

SURF descriptor is a 64 element vector. This vector is calculated in a domain oriented with the assigned angle and sized according to the scale of the feature. The descriptor is estimated using horizontal and vertical response histograms calculated in a 4 by 4 grid. There are two variants to this descriptor: the first provides a 32 element vector and the other one a 128 element vector. The algorithm uses integral images to implement the filters. This technique makes the algorithm very efficient.

The procedure to match SURF features is based on the descriptor associated with the extracted interest point. An interest point in the current image is compared with an interest point in the previous one by calculating a euclidean distance between their descriptor vectors.

3.2.3. Robust matching

A set of corresponding or matched points between two images are frequently used to calculate geometrical transformation models like affine transformations, homographies or the fundamental matrix in stereo systems. The matched points can be obtained by a variety of methods and the set of matched points usually has two error sources. The first is the measurement of the point position, which follows a gaussian distribution. The second is the *outliers* to the gaussian error distribution, which are the mismatched points given by the selected algorithm. These outliers can severely disturb the estimated function, and consequently alter any measurement or application based on this geometric transformation. The goal then, is to determine a way to select a set of *inliers* from the total set of correspondences, so that the desired projection model can be estimated with some standard methods, but employing only the set of pairs considered as inliers. This kind of calculation is considered as *robust estimation*, because the estimation is tolerant (robust) to measurements following a different or unmodeled error distribution (outliers).

Thus, the objective is to filter the total set of matched points in order to detect and eliminate erroneous matches, estimating the projection model employing only the correspondences considered as inliers. There are many algorithms

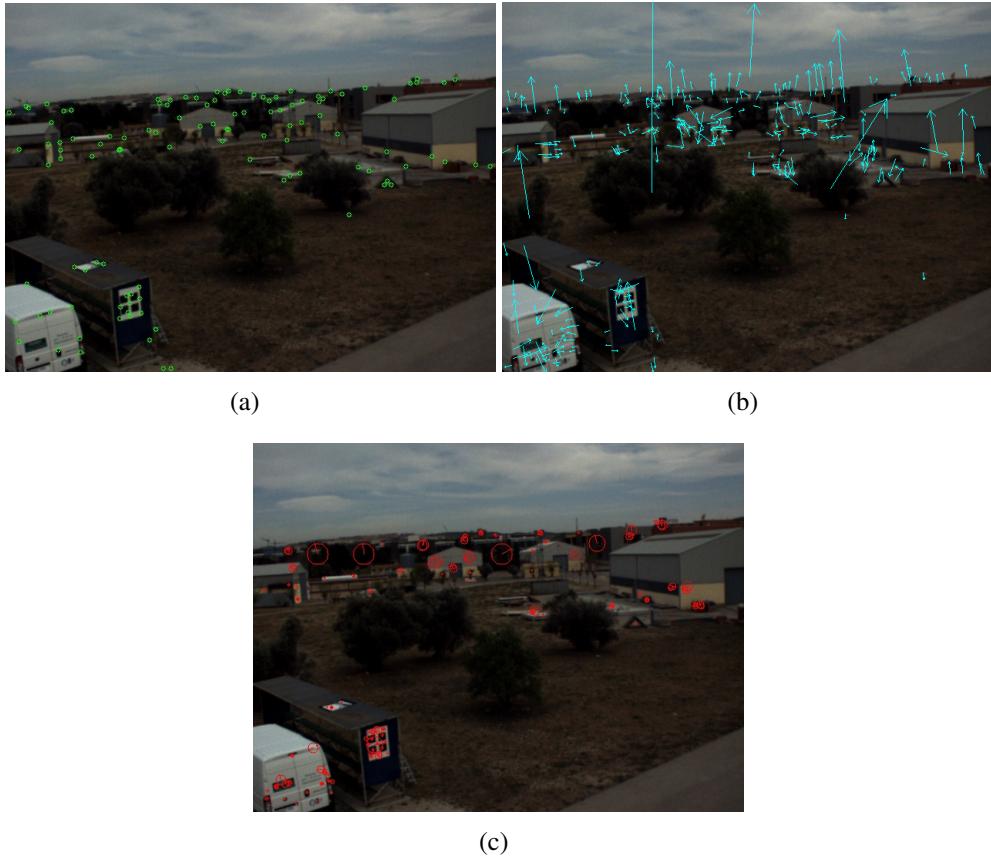


Figure 3.1: Comparison between features point extractors. Figure 3.1(a) are features obtained using Good Features to Track, Figure 3.1(b) are keypoints obtained using SIFT (the green arrows represents the keypoints orientation and scale) and Figure 3.1(c) are descriptors obtained using SURF (red circles and lines represents the descriptor scale and angle).

that have demonstrated good performance in model fitting, examples of which are the Median of Squares LMeds (Rousseeuw and Leroy, 1987) and Random Sample Consensus RANSAC algorithm (Fischer and Bolles, 1981). Both are randomized algorithms and are able to cope with a large proportion of outliers.

In order to use a robust estimation method for a projective transformation, we will assume that a set of matched points between two projective planes (two images) obtained using some of the methods describe above (sparse optical flow, SIFT, SURF, etc.) are available. This set includes some unknown proportion of outliers or bad correspondences so giving a series of matched points $(x_i, y_i) \leftrightarrow (x'_i, y'_i)$ for $i = 1 \dots n$, the robust estimation goal is to calculate a perspective transformation once the outliers have been discarded.

In order to discard the outliers from the set of matched points, the RANSAC algorithm is used. It achieves its goal by iteratively selecting a random subset

of the original data points by testing it to obtain the model and evaluating the model consensus, which is the total number of original data points that best fit the model. The model is obtained using a closed form solution according to the desired projective transformation (an example is show in the next subsection). This procedure is then repeated for fixed number of times, each time producing either a model which is rejected because too few points are classified as inliers, or a new refined model. When a maximum number of trials is reached, the algorithm return the projection model with the largest number of inliers. Algorithm 1 shows the general steps to obtain a robust transformation. Further description can be found on (Hartley and Zisserman, 2004) and (Fischer and Bolles, 1981).

Algorithm 1 Projective Transformation estimation using RANSAC

Require: Set of matched points $\mathbf{x}_i = (x_i, y_i) \leftrightarrow \mathbf{x}'_i = (x'_i, y'_i)$ for $i = 1 \dots n$

Define s = Minimum set of points to estimate the minimal solution.

Define p = Probability that at least one of the random samples is free from outliers

Define t = distance threshold to consider a point as an inlier for some model.

Define ε = Initial probability that any selected point is an outlier.

Define *Consensus* = Desired number of minimum Inliers based on the total number of matched points

Calculate the maximum number of samples $N = \log(1 - p) / \log(1 - (1 - \varepsilon)^s)$

while $N > Trials$ **do**

 Randomly select s pairs of matched points

 Calculate the minimal solution for the model under test, using selected s points

$inliers = 0$

for $i = 0$ to n **do**

 Calculate the distance $d_{transfer}^2 = d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2 + d(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}'_i)^2$

if $d_{transfer} < t$ **then**

$inliers = inliers + 1$

end if

end for

if $inliers > Consensus$ **then**

 Calculate the final projective transformation using all inliers points

$Consensus = inliers$

end if

 recalculate $\varepsilon = 1 - (inliers/n)$

 recalculate $N = \log(1 - p) / \log(1 - (1 - \varepsilon)^s)$

$Trials = Trials + 1$

end while

Robust Homography

Robust homography estimation employs the generic method described above. It can be viewed as the problem of estimating a 2D projective transformation, that given a set of points $\bar{\mathbf{x}}_i$ in \mathbb{P}^2 and a corresponding set of points $\bar{\mathbf{x}}'_i$ in \mathbb{P}^2 , compute the 3×3 matrix \mathbf{H} that takes each $\bar{\mathbf{x}}_i$ to $\bar{\mathbf{x}}'_i$ as is defined in Equation 3.13. In general the points $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}'_i$ are points in two images or in 2D plane surfaces.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.13)$$

$$\bar{\mathbf{x}}'_i = \mathbf{H} \cdot \bar{\mathbf{x}}_i$$

where: $(x, y, 1)^T$ are the homogeneous coordinates of the features in the reference image, and $(x', y', 1)^T$ are the homogeneous coordinates of the features $(x, y, 1)^T$, in the second image.

Taking into account that the number of degrees of freedom of the projective transformation is eight (defined up to scale) and because each point to point correspondences $(x_i, y_i) \leftrightarrow (x'_i, y'_i)$ give rise to two independent equations in the entries of \mathbf{H} , is enough with four correspondences to have an exact solution or minimal solution. If more than four points correspondences are given, the system is over-determined and \mathbf{H} is estimated using a minimization method. So, in order to use the algorithm 1, we define the minimum set of points to be $s = 4$.

If matrix \mathbf{H} is written in the form of a vector $\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^t$, the homogeneous equations $\bar{\mathbf{x}}' = \mathbf{H}\bar{\mathbf{x}}$ for n points could be formed as $\mathbf{Ah} = 0$, with \mathbf{A} a $2n \times 9$ matrix defined by Equation 3.14:

$$A = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\ \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n & -y'_n \end{bmatrix} \quad (3.14)$$

In general, Equation 3.14 can be solved using three different methods (the inhomogeneous solution, the homogeneous solution and non-linear geometric solution) as explained in (Criminisi et al., 1999):

- **Inhomogeneous solution:** In this method, one of the nine matrix elements is given a fixed unity value, forming an equation of the form $\mathbf{A}'\mathbf{h}' = \mathbf{b}$ as

shown in Equation 3.15

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ \vdots \\ x'_n \\ y'_n \end{bmatrix} \quad (3.15)$$

The resulting simultaneous equations for the 8 unknown elements are then solved using a gaussian elimination in the case of a minimal solution or using a pseudo-inverse method in case of an over-determined system (Hartley and Zisserman, 2004). Last one is the most commonly used method.

- **Homogeneous solution:** The solution is obtained using Singular Value Decomposition SVD, taking into account that the vector \mathbf{h} that minimizes the algebraic residuals \mathbf{Ah} , subject to $\|\mathbf{h}\| = 1$, is given by the eigenvector of least $\mathbf{A}^t\mathbf{A}$. This eigenvector can be obtained directly from the SVD of \mathbf{A} . In the case of $n = 4$, \mathbf{h} is the null-vector of \mathbf{A} and the residuals are zero.
- **Non-linear geometric solution:** The summed euclidean distances between the measured and a mapped point are minimized. This method has the advantage, over the above two algebraic methods, that the quantity minimized is meaningful and corresponds to the error involved in the measurement. There is no closed form solution.

Figure 3.2 shows an example of feature matching on images taken using a UAV, in which SURF algorithm, is used to obtain visual features , and the RANSAC algorithm is used for outliers rejection.

3.3. 3D estimation based on Homographies

In this section, a 3D pose estimation method based on the analysis of the camera projection matrix and the homography model is presented. The method estimates the position of a world plane relative to the camera projection center for every image sequence using previous frame-to-frame homographies and the projective transformation at first frame, obtaining for each new image, the camera rotation matrix \mathbf{R} and a translational vector \mathbf{t} . This method is based on the one propose by Simon *et. al.* (Simon et al., 2000), (Simon and Berger, 2002).

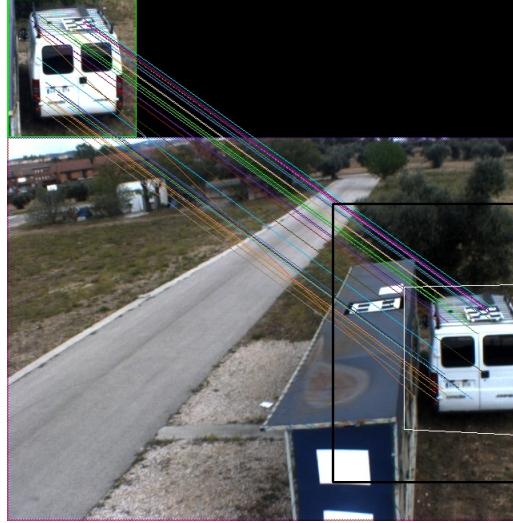


Figure 3.2: Robust homography estimation using SURF features on a car tracking from a UAV. Up: reference template. Down: scene view, in which translation, rotation, and occlusions are present.

3.3.1. World plane projection onto the Image plane

In order to align the planar object in the world space and the camera axis system, we consider the general pinhole camera model and a homogeneous camera projection matrix, that maps a world point \mathbf{x}_w in \mathbb{P}^3 (projective space) to a point \mathbf{x}^i on i^{th} image in \mathbb{P}^2 , defined by Equation 3.16:

$$s\mathbf{x}^i = \mathbf{P}^i \mathbf{x}_w = \mathbf{K} [\mathbf{R}^i | \mathbf{t}^i] \mathbf{x}_w = \mathbf{K} [\mathbf{r}_1^i \quad \mathbf{r}_2^i \quad \mathbf{r}_3^i \quad \mathbf{t}^i] \mathbf{x}_w \quad (3.16)$$

where the matrix \mathbf{K} is the camera calibration matrix, \mathbf{R}^i and \mathbf{t}^i are the rotation and translation that relates the world coordinate system and camera coordinate system, and s is an arbitrary scale factor. Figure 3.3 shows the relation between a world reference plane and two images taken by a moving camera, showing the homography induced by a plane between these two frames.

If point \mathbf{x}_w is restricted to lie on a plane Π , with a coordinate system selected in such a way that the plane equation of Π is $Z = 0$, the camera projection matrix (Equation 3.16) can be written as Equation 3.17:

$$s\mathbf{x}^i = \mathbf{P}^i \mathbf{x}_{\Pi} = \mathbf{P}^i \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \langle \mathbf{P}^i \rangle \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3.17)$$

where $\langle \mathbf{P}^i \rangle$ denotes that this matrix is deprived on its third column or $\langle \mathbf{P}^i \rangle = \mathbf{K} [\mathbf{r}_1^i \quad \mathbf{r}_2^i \quad \mathbf{t}^i]$. The deprived camera projection matrix is a 3×3

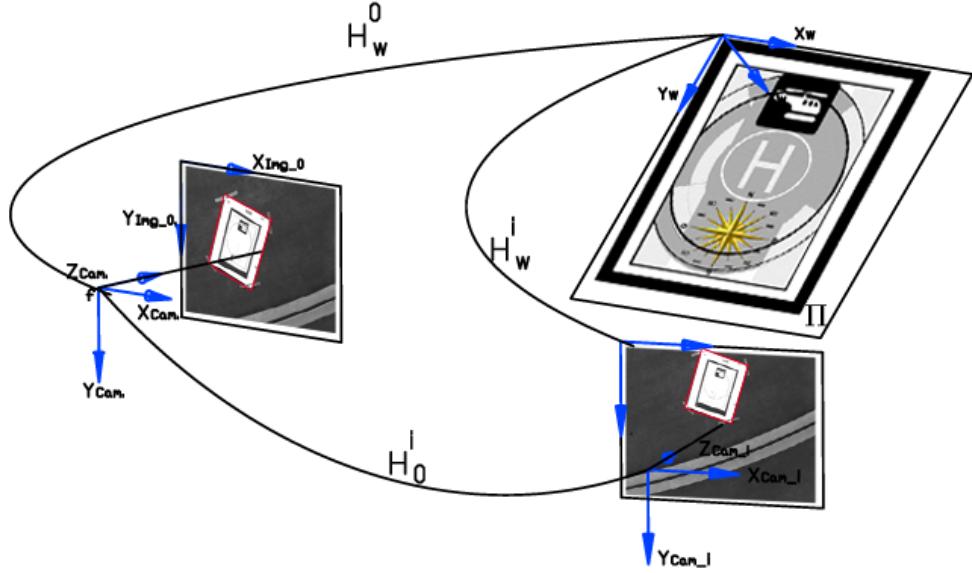


Figure 3.3: Projection model on a moving camera and frame-to-frame homography induced by a plane.

projection matrix, which transforms points in the world plane (now in \mathbb{P}^2) to the i^{th} image plane (likewise in \mathbb{P}^2), that is none other than a planar homography \mathbf{H}_w^i defined up to scale factor as Equation 3.18 shows.

$$\mathbf{H}_w^i = \mathbf{K} [\mathbf{r}_1^i \quad \mathbf{r}_2^i \quad \mathbf{t}^i] = \langle \mathbf{P}^i \rangle \quad (3.18)$$

Equation 3.18 defines the homography which transforms points in the world plane to the i^{th} image plane. Any point in the world plane $\mathbf{x}_\Pi = [x_\Pi, y_\Pi, 1]^T$ is projected onto the image plane as $\mathbf{x} = [x, y, 1]^T$. Because the world plane coordinates system is not known for the i^{th} image, \mathbf{H}_w^i can not be directly evaluated. However, if the position of the world plane for a reference image is known, a homography \mathbf{H}_w^0 , can be defined. Then, the i^{th} image can be related with the reference image to obtain the homography \mathbf{H}_0^i . This mapping is obtained using sequential frame-to-frame homographies \mathbf{H}_{i-1}^i , calculated for any pair of frames ($i-1, i$) and used to relate the i^{th} frame to the first image \mathbf{H}_0^i using Equation 3.19:

$$\mathbf{H}_0^i = \mathbf{H}_{i-1}^i \mathbf{H}_{i-2}^{i-1} \cdots \mathbf{H}_0^1 \quad (3.19)$$

This mapping and the aligning between the initial frame to the world plane reference is used to obtain the projection matrix between the world plane and the i^{th} image \mathbf{H}_w^i as shown in Equation 3.20:

$$\mathbf{H}_w^i = \mathbf{H}_0^i \mathbf{H}_w^0 \quad (3.20)$$

In order to apply Equation 3.20, to relate the world plane and the i^{th} image, we must know the homography \mathbf{H}_w^0 . A method to obtain it, needs a user that selects four points in the image that corresponds to four points on a rectangle in the scene, forming the matched points $(0,0) \leftrightarrow (x_1,y_1)$, $(0,\Pi_{Width}) \leftrightarrow (x_2,y_2)$, $(\Pi_{Length},0) \leftrightarrow (x_3,y_3)$ and $(\Pi_{Length},\Pi_{Width}) \leftrightarrow (x_4,y_4)$. Alternative, the selection of the four point in the image plane can be done by using a rectangle segmentation method on the image plane. The four points selection generates a world plane defined in a coordinate system in which the plane equation of Π is $Z = 0$. With these four correspondences between the world plane and the image plane, the minimal solution for homography $\mathbf{H}'_w^0 = [\mathbf{h}'_1^0 \ \mathbf{h}'_2^0 \ \mathbf{h}'_3^0]$ is obtained using some of the methods (e.g. Inhomogeneous solution) described in section 3.2.3. Then, the aspect ratio of the world rectangle ($s_{w.r.}$), obtained from \mathbf{H}'_w^0 is used to calculate the corrected homography \mathbf{H}_w^0 using Equation 3.21

$$\mathbf{H}_w^0 = \mathbf{H}'_w^0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/s_{w.r.} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{where} \quad (3.21)$$

$$s_{w.r.} = \frac{\|\mathbf{K}^{-1}\mathbf{h}'_2^0\|}{\|\mathbf{K}^{-1}\mathbf{h}'_1^0\|}$$

The planar homographies \mathbf{H}_w^i , could be divided in metric and projective components. The metric components are the selected euclidean coordinates in the plane and the scale reconstruction (obtained using scene scale information). The projective component concerns the orthogonality of the coordinate axes. The rotation matrix and the translation vector are computed from the plane to image homography using the method described in (Zhang, 2000).

From Equation 3.18 and defining the scale factor $\lambda = 1/s$, we have that:

$$[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] = \lambda \mathbf{K}^{-1} \mathbf{H}_w^i = \lambda \mathbf{K}^{-1} [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] \quad \text{where} \quad (3.22)$$

$$\mathbf{r}_1 = \lambda \mathbf{K}^{-1} \mathbf{h}_1, \quad \mathbf{r}_2 = \lambda \mathbf{K}^{-1} \mathbf{h}_2, \quad \mathbf{t} = \lambda \mathbf{K}^{-1} \mathbf{h}_3$$

The scale factor λ can be calculated using Equation 3.23:

$$\lambda = \frac{1}{\|\mathbf{K}^{-1} \mathbf{h}_1\|} = \frac{1}{\|\mathbf{K}^{-1} \mathbf{h}_2\|} \quad (3.23)$$

Because the columns of the rotation matrix must be orthonormal, the third vector of the rotation matrix \mathbf{r}_3 could be determined by the cross product of $\mathbf{r}_1 \times \mathbf{r}_2$. However, the noise of the homography estimation causes that the resulting matrix $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]$ does not satisfy the orthonormality condition and we must find a new rotation matrix \mathbf{R}' that best approximates to the given matrix \mathbf{R} according to smallest Frobenius norm for matrices (the root of the sum of squared matrix coefficients) (Sturm, 2000), (Zhang, 2000). As demonstrated by (Zhang, 2000), this problem can be solved by forming the rotation matrix $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]$ and using singular value decomposition SVD to form the new optimal rotation matrix \mathbf{R}' as Equation 3.24 shows:

$$\begin{aligned}\mathbf{R} &= [\mathbf{r}_1 \ \mathbf{r}_2 \ (\mathbf{r}_1 \times \mathbf{r}_2)] = \mathbf{U}\mathbf{S}\mathbf{V}^T \\ \mathbf{S} &= \text{diag}(\sigma_1, \sigma_2, \sigma_3) \\ \mathbf{R}' &= \mathbf{U}\mathbf{V}^T\end{aligned}\tag{3.24}$$

Thus, the solution for the camera pose problem is defined by Equation 3.25:

$$\mathbf{x}^i = \mathbf{P}^i \mathbf{X} = \mathbf{K}[\mathbf{R}'|\mathbf{t}] \mathbf{X}\tag{3.25}$$

The proposed method has been implemented using C/C++ libraries. Optical flow uses the OpenCV functions (Bradski and Kaehler, 2008), (Bradski, 2000) while ICIA uses the implementation presented in (Buenaposada et al., 2003). The algorithm has been implemented and tested on the Via nano-ITX computer on-board the Colibri 3 UAV (Appendix B.1). It is a single core 1.5 GHz processor with 2.0 GB of memory. The proposed method runs on this computer with an average of 12 fps on 640x480 pixels color/monochrome images captured with a firewire camera and a 6 mm lens.

3.4. 3D pose performance evaluation

The 3D pose estimation method presented above, has been experimentally validated against a ground truth system composed of a VICON MX vision-based motion capture system (Vicon Motion Systems, 2011)¹ The employed Vicon system is made up of 5 infrared T40 cameras capable of capturing 370 frames per second at a full frame resolution of 4 Megapixels, a MX Giganet hub and the Vicon Tracker software. The system is capable of multiple, accurate and real time (latency of 2.5 milliseconds) object tracking within a defined

¹The results presented on this section have been obtained in part during a research stay on the Australian Research Centre for Aerospace Automation ARCAA (ARCAA et al., 2010) between July and October 2010, as part of the staff interchange program under the International Cooperation Program for Unmanned Aerial Systems Research and Development ICPUAS (ICPUAS et al., 2009)).

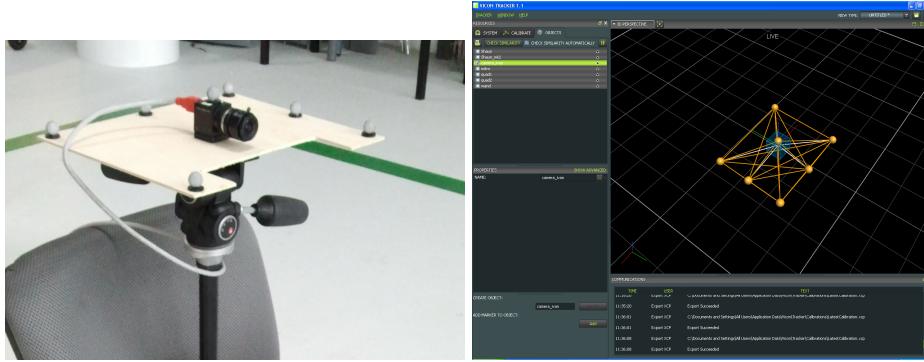


Figure 3.4: Vicon Camera model using for testing the performance of the 3D pose estimation algorithm: Left is the camera with the attached IR marks. Right the camera model captured by the Vicon System.

workspace. Object tracking is based on a series of infrared IR marks, forming a desired pattern in the workspace. The system can provide information like 3D pose with sub-millimeter precision (axes X , Y and Z) and sub-degree for attitude.

Considering the performance evaluation tests, a calibrated firewire camera with a resolution of 1024x740 pixels and a 8 mm lens capturing images at 7.5 fps has been used. A series of IR marks were attached to the camera body frame in order to generate a Vicon captured object, as shown in Figure 3.4. The working space is an area of 7.9x5.7 m and height of 2 m. In this area, the Vicon system employs at least 3 of the 5 available cameras to estimate the 3D pose and attitude of the captured object. The relation between the camera and IR marks are known. It is used for rotating the image data in order to compare it with the reference system. The VICON workspace is shown in Figure 3.5.

A series of tests were performed in order to acquire diverse sets of images, each one with the associated 3D pose in the working space and the attitude data. The capture process is synchronized with the VICON hub throughout a TCP/IP server.

3.4.1. 3D pose tests and results

The proposed 3D estimation method based on landmarks was evaluated by using a scaled helipad and the calibrated camera system within the Vicon workspace as shown in Figure 3.7. The helipad was selected according to the proposed method and the camera moved in the workspace while the helipad is tracked. The estimated helipad position (with respect camera) and the camera position in the Vicon workspace are stored and associated to each one of the

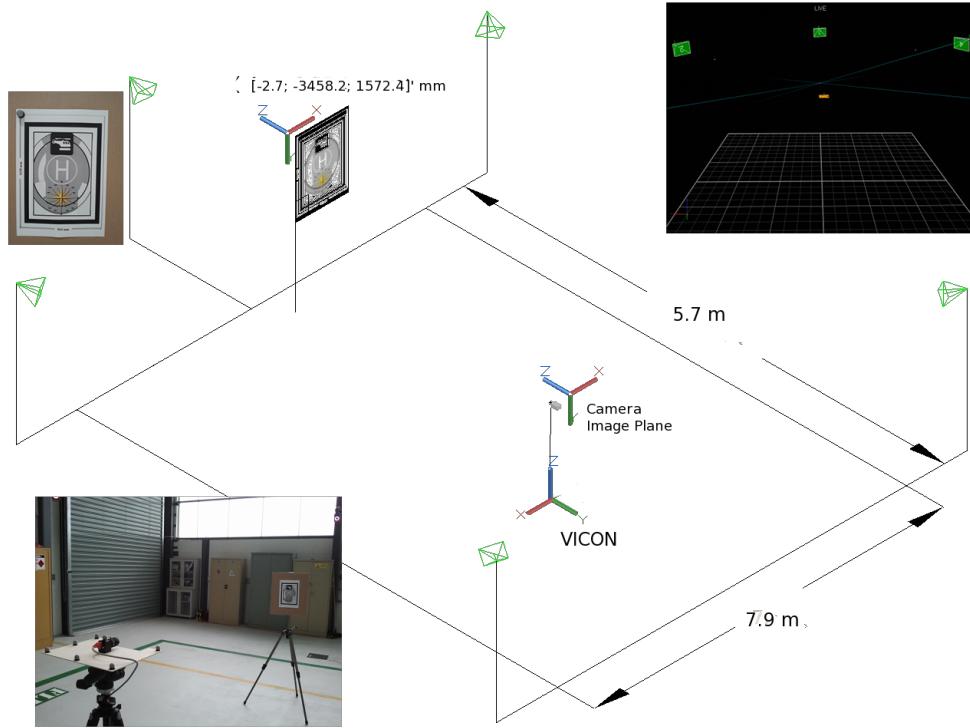


Figure 3.5: Vicon workspace for testing the performance of the proposed algorithms. Left-down is the camera in the middle of the workspace. Left-up: Is the plane used as reference with one IR mark. Right camera and plane captured by the Vicon System.



Figure 3.6: 3D pose estimation based on landmarks performance test: Left is the scaled helipad and the camera on the Vicon workspace. Right the helipad imaged captured by the camera with the associated estimation.

captured images sequences. This data is used for performance evaluation.

Table 3.1: 3D pose performance tests

Test	Vicon axes	Image axes
Test 1	X	X (lateral displacement)
Test 2	Y	Z (depth change)
Test 3	Z	Y (vertical displacement)
Test 4	X, Y+Yaw Rot.	X, Z (lateral and depth Trans.) + Pitch Rot.

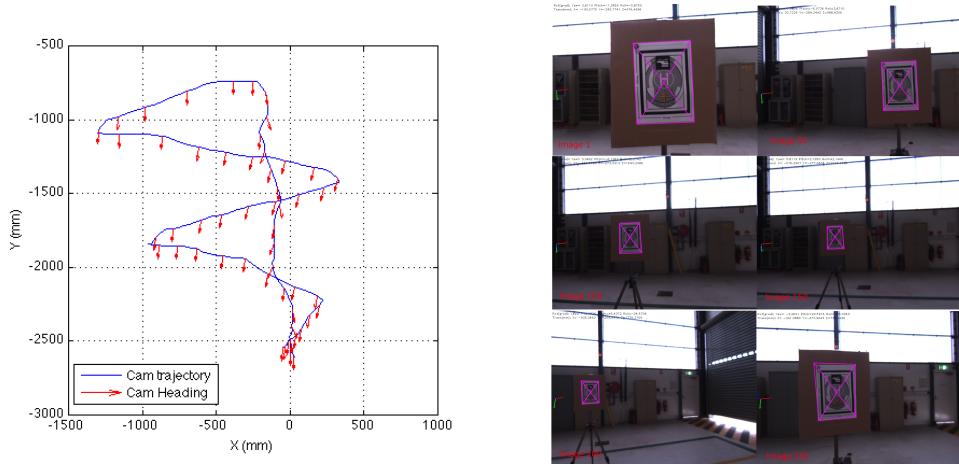


Figure 3.7: 2D trajectory and camera heading reconstruction for test 4. Left: 2D reconstruction using Vicon data. Right: Images 1, 50, 100, 150, 200 and 250 with the estimated pose.

Three different tests were performed, in which the camera is manually moved parallel to one axis in the Vicon workspace. They have been performed by applying camera translations in X, Y and Z axes. A last test was performed in which the camera is moved on a 2D plane making lateral and depth movements w.r.t the reference landmark. For all test camera have been maintained level0.3, therefore not important *Pitch* and *Roll* rotations are applied to the system. Special interest is to evaluate the depth estimation, that in these tests correspond to changes in Y axis. Table 3.1 describes the tests that have been performed, explaining how these movements are reflected in the image coordinate system.

Figure 3.7 shows the reconstruction of the 2D camera trajectory using the Vicon data for Test 4. It also shows some snapshots of the estimation process based on homographies for this test.

Figures 3.8 to 3.10, shows the results of the 3D homography estimation method compared with the VICON reference system. In this tests a landmark with a size of 210x297 mm was employed in order to get the scale dimension to

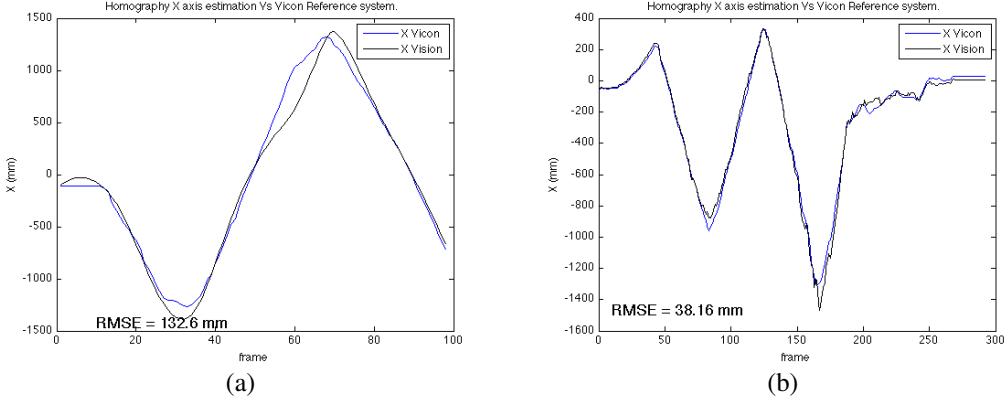


Figure 3.8: Comparison between the X axis displacement for homography estimation and VICON reference system. Movements parallel to this axis corresponds to lateral displacements in image plane. Average frame rate is 7.5 Hz. (a) Test 1. (b) Test 4.

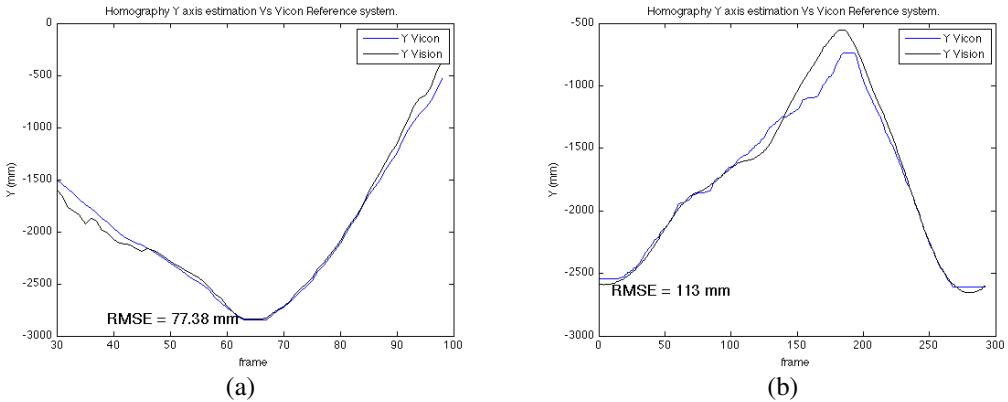


Figure 3.9: Comparison between the Y axis displacement for homography estimation and VICON reference system. Movements parallel to this axis corresponds to scale changes on the image plane. Average frame rate is 7.5 Hz. (a) Test 2. (b) Test 4.

the pose estimated by the homography decomposition. Because the landmark has attached a IR mark on the up-left corner, relation between the landmark and the Vicon coordinate systems is known ($\mathbf{T}_L = [-2.7, 3458.2, 1572.4]^T$ mm). It is used for data comparison.

Figure 3.8 show the response for the camera displacement compared with Vicon X axis for tests 1 and 4. Figure 3.9 shows the comparison for Vicon Y axis for tests 2 and 4. Figure 3.10 corresponds to a camera altitude change in the Vicon Z axis for test 1 and 3.

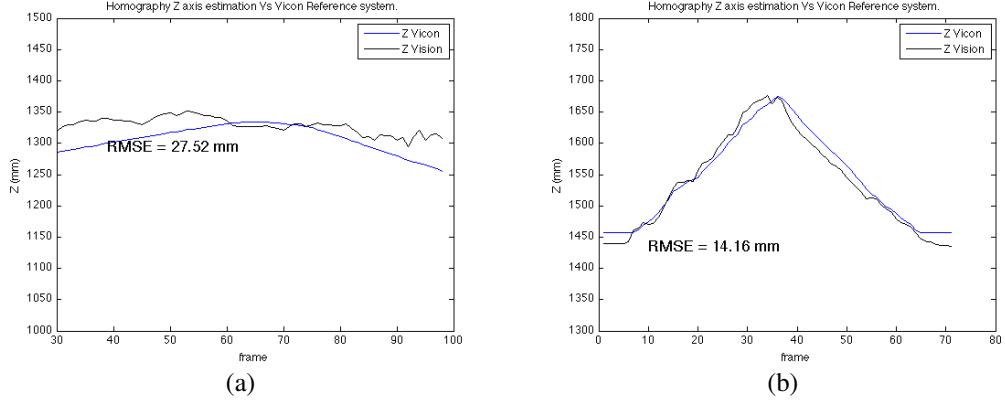


Figure 3.10: Comparison between the Z axis displacement for homography estimation and VICON reference system. Movements parallel to this axis corresponds to vertical displacements in image plane. Average frame rate is 7.5 Hz. (a) Test 1. (b) Test 3.

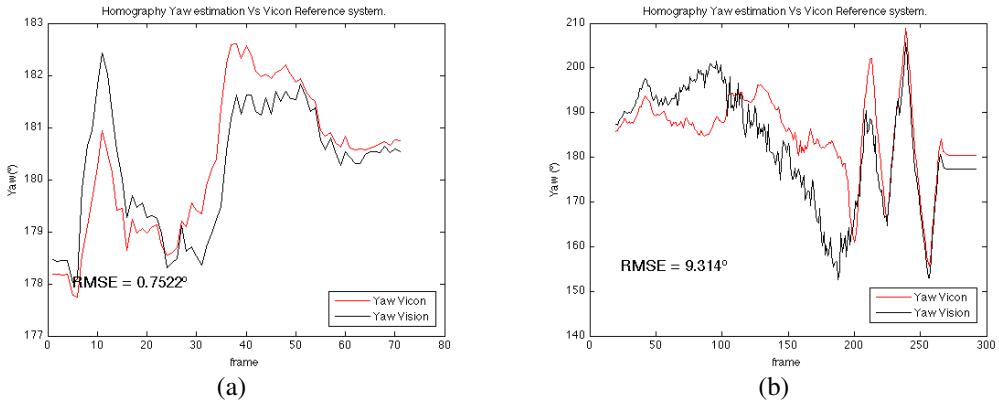


Figure 3.11: Comparison between the Yaw rotation for homography estimation and VICON reference system. Yaw angle corresponds to a rotation in Z axis in Vicon system and a negative rotation in Y axis on image coordinate system. Average frame rate is 7.5 Hz. (a) Test 3. (b) Test 4.

Finally, Figure 3.11 shows the comparison between homography Yaw estimation and Vicon Yaw for tests 3 and 4. In these tests Yaw angle corresponds to a negative rotation in Y axis (*Pitch*) on image coordinate system.

Result have shown that the estimated data is coherent with the measured data given by the VICON system. The Root Mean Square Error RMSE for all tests are below 0.3 m in a working range of 5 m, given a good precision for pose estimation, in which classical systems have worst performance (i.e GPS

resolution). Yaw estimations also are coherent with ground truth data. Pose estimations depends of the quality of calculated homography. In this tests there are accumulative errors caused by the concatenations of homographies between consecutive images. Absolute homographies between reference and current image can also be used for estimating relative position. However, homography estimation method must be robust against large displacements and scale changes.

3.5. Conclusions

This chapter has presented a review of the visual processing techniques and algorithms used on the development of this thesis. These techniques include visual algorithms for features detection and tracking based on features and/or correlation. These methods are employed to robustly obtain a projective transformation.

A method to estimate a camera 3D position and orientation based on planar landmarks is presented. The method based on the analysis of a projective transformation and the general pinhole model, allows to reconstruct the metric position and attitude of a planar landmark w.r.t the camera frame. This method uses computer vision techniques in order to give a reliable metric measurement of the camera movement in the workspace.

The precision of the proposed system has been tested in a controlled working space and evaluated against a ground truth reference given by a motion capture system. Results have shown that the system is coherent with the ground truth data and the obtained results have enough accuracy for 3D pose applications based on visual information.

This chapter contributes to the aim of this thesis by solving the problem of measuring the 3D position of a moving camera based only on visual information that it provides. The method has demonstrated to be robust against the camera moving and scale factor changes, as well as fast enough to be implemented on the low computational performance units common in light UAVs.

Chapter 4

UAVs on-board visual odometry and video enhancement

4.1. Introduction

This chapter presents the uses of visual tracking and robust projective model estimation techniques to solve the problem of UAVs on-board pose estimation, as well as their use for video enhancement. First, a video stabilization algorithm used to reduce the effect of vibrations and movements on an image is presented. Then, a method for mosaic building is presented based on the alignment of images related by a projective model. Finally a method for UAV pose estimation based on homographies is presented.

4.2. Aerial Video Stabilization

This section introduces a system for video stabilization used on vision-based UAV control tasks. It is composed of three stages. The first stage involves the matching of features between consecutive images, using previously mentioned methods (Lucas Kanade optical flow, SIFT and SURF features). In the second stage, the camera motion is robustly estimated by fitting a projective motion model using the matched features and the RANSAC algorithm (Fischer and

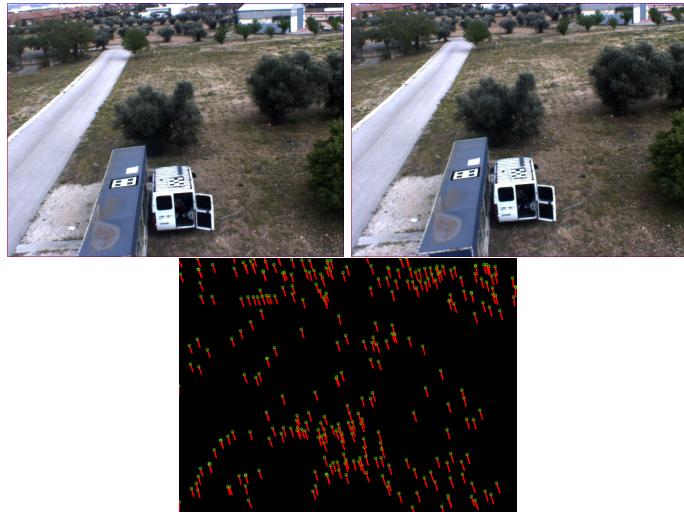


Figure 4.1: Optical flow between consecutive images using the p pyramidal Lucas Kanade Tracker. Up: two frames with high translational movement. Down: optical flow between frames. Red line is the translational movement and green circle is the final position of the feature.

Bolles, 1981). The last stage, a transformation between the original video and the stabilized sequence is obtained employing the estimated projection model.

■ Image feature matching

On images with high motion, good matched features can be obtained using the pyramidal Lucas Kanade algorithm modification (Bouguet Jean Yves, 1999). It is used to solve the problem that arise when large and non-coherent motion are present between consecutive frames, by first tracking features over large spatial scales on the pyramid image, obtaining an initial motion estimation, and then refine it by down sampling the levels of the images pyramid until it arrives to the original scale. Figure 4.1 shows the matched features or the optical flow obtained using the pyramidal Lucas Kanade tracker on two consecutive images obtained during a UAV flight.

Stabilized images for UAV often are required to be on real time. Because CPU based implementations of LKT for high resolution are not fast enough, new approaches based on the computational power of the graphics processing unit GPU present in modern computers hardware, have become a key factor to give a great acceleration for computer vision algorithms. In the case of the LKT, the GPU implementation increases the computational speed up to 15 times faster than their optimized CPU counterparts, allowing real time processing algorithms for high resolution images (Sinha et al., 2006), (Zach et al., 2008).

SIFT or SURF keypoints can be used instead of LKT optical flow, increasing the robustness to scale and illumination changes. The object is matched along

the image sequence comparing the model template and the descriptor of the current image, using the nearest neighbor method improved with the Kd-tree search algorithm and the Best Bin First search modification (Beis and Lowe, 1997). However, the features estimation and matching have a big computational cost, making a real time implementation difficult when applied to a full size input image.

■ Robust camera motion estimation

Matched features are used to obtain a motion defined in Equation 4.1 using RANSAC adjustment algorithm. The method is very similar to the one described in section 3.2.3, with the difference that a euclidean, affine or homography model is estimated according to the desired projection model.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.1)$$

where: $(x, y)^T$ are the homogeneous coordinates of the features in the k^{th} image I_k , (x', y') are the homogeneous coordinates of the features $(x, y)^T$, in the $k + 1^{th}$ image I_{k+1} . Parameters h_{ij} depends of the selected model. For the homography model the parameters corresponds to the eight degree of freedom of the model with $h_{33} = 1$. For the affine model the parameters h_{11}, h_{12}, h_{21} and h_{22} control the scaling and two rotations, while parameters h_{13} and h_{23} correspond with the translation vector. For the euclidean model the parameters $h_{11} = \cos(\theta)$, $h_{12} = -\sin(\theta)$, $h_{21} = \sin(\theta)$ and $h_{22} = \cos(\theta)$ control the image plane rotation, while parameters h_{13} and h_{23} correspond with the translation vector.

In the case of the homography, the algorithm, iteratively selects a random subset of four of the original data points, forming for each one two set of equations and solving the linear system of the form $\mathbf{Ah} = \mathbf{b}$ shown in Equation (4.2) in order to get a homography projection (solved using a gaussian elimination in the case of a minimal solution or using a pseudo-inverse method in case of an over-determined system). Then, it is tested for all original points, obtaining the trial model consensus (total number of original data points that fit the transformation model). This procedure is repeated by a defined number of trials (reduced when the consensus increases), each time producing either a rejected model with few inliers, or a new model with a best consensus. When the total trials are reached, the model with the best consensus is selected. A similar approach can be used to estimate the euclidean and affine models using a minimal set of three matched features on each trial, and solving the respective linear system.

.



Figure 4.2: Video Stabilization of image I_{k+1} with respect to the reference image on the sequence. Red Box: original image. Left: current frame without stabilization. Right: stabilized frame with respect to the reference image.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_n \\ y'_n \end{bmatrix} \quad (4.2)$$

■ Stabilized image generation

The video stabilization takes a specific image on the sequence as reference. Generally, this is done with respect to the first image (but not restricted to this). The cartesian product of successively frame projective transformations ($I_{k+1}(x, y) = \mathbf{H}_{1 \rightarrow k+1} \cdot I_1(x, y) = (\prod_{j=1}^k \mathbf{H}_{j \rightarrow j+1}) \cdot I_1(x, y)$) is employed in order to obtain the affine transformation between the reference image and the current frame. Back-projecting each new image I_{k+1} with the inverse of the cartesian projection product, the stabilized image sequence is obtained. Figure 4.2 shows an example of the on-board image stabilization test with respect to the original frame. Tests have shown good performance and robustness in dynamic scenes with big image translation (up to 300 pixels) and illumination changes. The stabilization process is performed on images of 640x480 pixels with an average frequency 15 fps.

If the GPU-LKT (Sinha et al., 2006) is employed to obtain the optical flow (Robust homography is still estimated on the CPU using the GPU points), the performance of the algorithm increases significantly, making possible to process images of 1024x780 pixels obtaining 1000 features distributed over all image, with a frequency of 25 fps on a GPU of 16 cores. Figure 4.3 shows an example of a real time video stabilization on images sequences (1024x780 pixels) taken

with a UAV and performed using a GPU KLT tracker.

Figure 4.4 shows a comparison between the stabilization process depending of the selected projective transformation (affine, euclidean and homography). This test was done using the GPU KLT implementation, allowing to have a 1000 points tracking at a frame rate of 25 Hz on 720x480 colored images. The stabilization result (human eye appreciation) varies according to the selected method. In the case of euclidean transformation, the stabilized images do not preserve the scale changes, making it ideal for not zooming images with lateral movements. Meanwhile, the homography model correctly drives zooming and scale changes, but deformations and drift problems can be notorious on large camera and UAV displacements.

Additional test and stabilized video sequences can be seen on the Vision4UAV project web page (CVG-UPM, 2010).

4.3. On-board mosaic building

Mosaic building is the process of generate a large image or composition based on the alignment of multiples spare images that are generally related by a projective transformation. The presented method is valid for images that can be registered by means of a homography. As in the case of image stabilization, mosaics are constructed from a defined reference frame. Usually, this reference frame corresponds to any one of the source images. However in real time mosaicing on image sequences, the first is usually selected as the reference image for the building process. A basic mosaicing process involves three steps: The registration process, the image reprojection and the blending function (Capel, 2001). Following is a review of this 3 step process focused on video sequences mosaicing:

- The registration process is in essence the same process explained above for video stabilization, in order to obtain for each image I_{k+1} , the projective transformation $\mathbf{H}_{1 \rightarrow k+1}$ that relates it with the reference image I_1 . This process can be done either with a direct registration between I_1 and I_{k+1} or using a cartesian product of successively frame projective transformations ($I_{k+1} = (\prod_{j=1}^k \mathbf{H}_{j \rightarrow j+1}) \cdot I_1$).
- The reprojection involves two stages, the first one is a one-to-one mapping between each pixel of each image onto a point on the global frame (the reference frame) using the inverse of the projective transformation

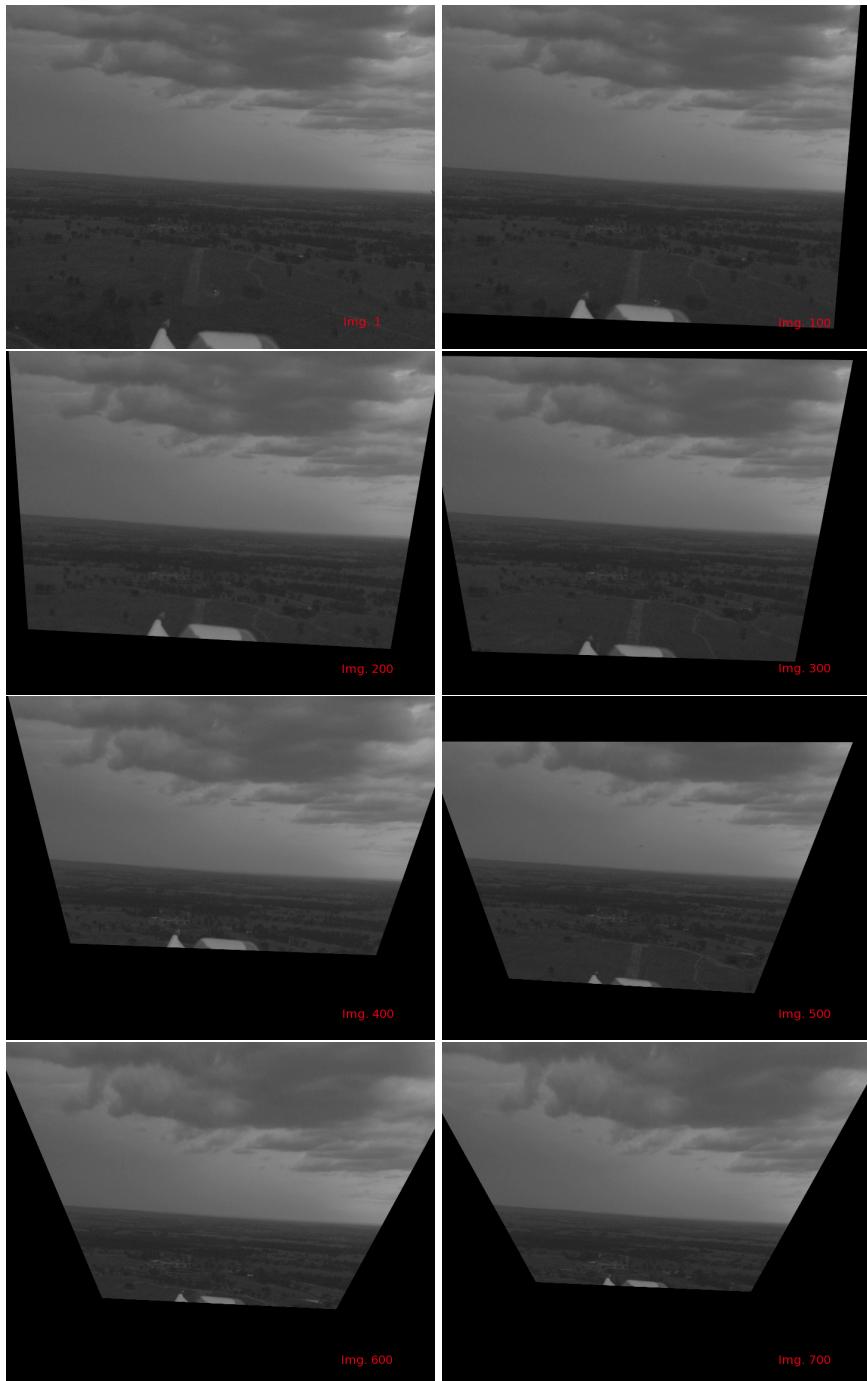


Figure 4.3: Real time (25 fps) Video Stabilization on a UAV sequence of 700 images (1024x780 pixels) using a GPU LKT implementation. The tracker estimates the optical flow of 1000 points, and then, a robust homography is estimated in order to make the stabilization process. Showed images correspond to frames 100, 200, 300, 400, 500, 600 and 700. Images courtesy of ARCAA, were taken on a flight test using the Flamingo UAV (ARCAA et al., 2010).

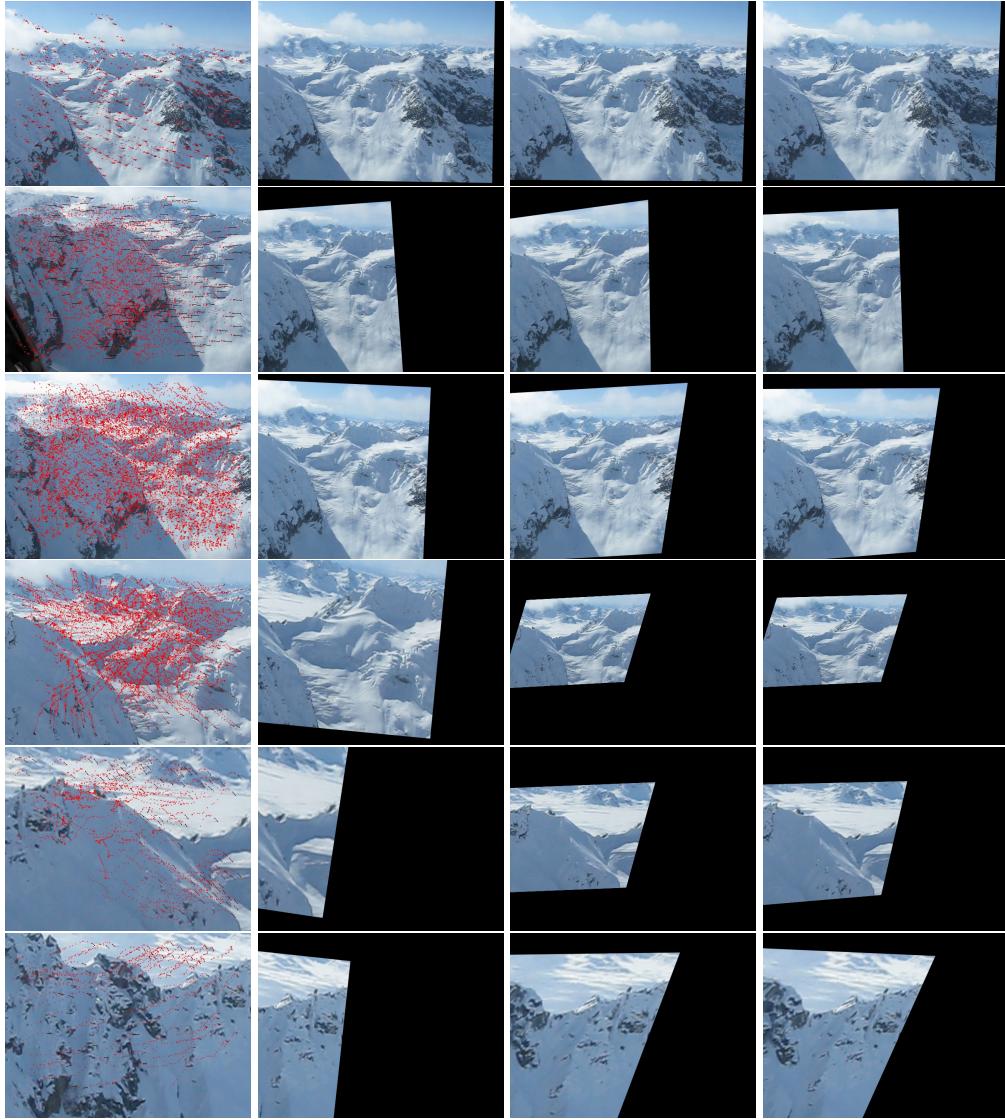


Figure 4.4: Real time (25 fps) Video Stabilization on a sequence of 500 images using a GPU LKT implementation over 1000 points. First Images column corresponds to the originals captured images with the tracked points represented in red. Second column correspond to stabilization using a euclidean projection model, third column is a stabilization using an affine projection model and last column is using a homography. Stabilized frames are show in images 10, 50,100,150,200, and 250.

$\mathbf{H}_{1 \rightarrow k+1}$, forming the *mosaic representation* on the reference frame. The projected point on the global frame is then reprojected onto the rendered image (viewed image) through a mapping function called *rendering transformation*. The most simple mapping function is a similarity transformation (scaling and translation) sometimes called as a *planar rendering*. It allows generating a viewable scaled image of the mosaic

representation and shifting the origin of the global reference frame to the origin of the rendered imaging. There are also others rendering transformations that use rotation, cylindrical or spherical projections (Capel, 2001) that depends of the desired properties of the rendered image (the image sensor in the virtual camera). A planar rendering emulates the view obtained from an ideally perspective camera with a large sensor and wide field of view lens (not radial distortion). It is used when the motion is a perfect pan movement or when the camera moves parallel to a pure planar and distant scene. Planar rendering preserves straight lines and perspective projection. On the other hand, a cylindrical projection tries to emulate a panoramic camera, in which the film forms a cylindrical surface around the focal point. It is used when there is a pure rotation movement (perfect pan) around the focal point. In order to generate the projection into the cylindric surface it is necessary to know the camera intrinsic parameters. The cylindric projection is very useful for 360° pure rotations panoramas. However, straight line and perspective projection are not preserved on the cylindric rendered image. A similar behavior presents a spherical projection.

- The blending function is used to drive the interpolation process given by the projection onto the rendered image of several pixels of multiple source images according to their respective overlapping proportions. There are several ways to drive the blending process, e.g. functions based on averaging, feathering or nearest image center among others. In the averaging method, the output pixel value is obtained as the simple average of all the pixels that are projected on it. On the feathered blending, a weighting function is applied to each image in order to give more importance to the centre of the source image and reduce it to zero on the image boundary. The interpolated pixel is then computed from the average of the weighted values extracted from the source images. A common blending function is a quadratic filter $f(u, v) = (1 - (u - \frac{\text{ImgWidth}}{2})^2)(1 - (v - \frac{\text{ImgHeight}}{2})^2)$. Finally, in the nearest image centre, the distance of each pixel to its source image centre is computed. For each point on the mosaic image, a list of source pixels projected on it and ranked according to the measured distance is generated. The pixel closest to its image center is then selected as the output value.

Figure 4.5, summarizes the mosaic process for a video sequence. Often, the reprojection and blending process are fused on a unique step, forming a mapping function made up of a similarity transformation on a weighted perspective projection for each pixel on the source images.

The process depicted above has been applied on different images sequences obtained from UAVs and hand held cameras. In each test, the first image of the

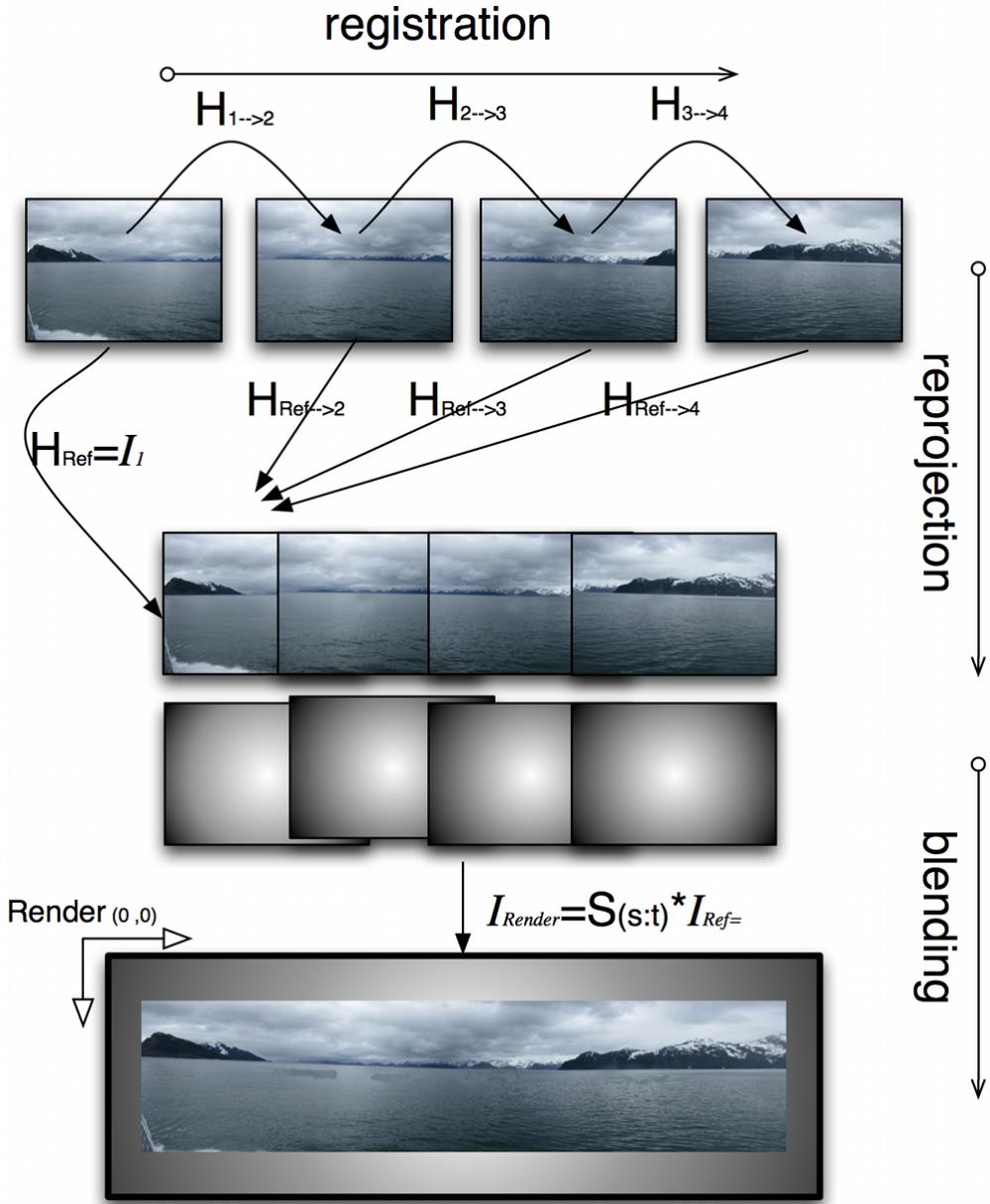


Figure 4.5: Mosaic building process. Image 1 is used as a reference to re-project the successive images using the homography projection $H_{1 \rightarrow k+1}$. The source images and pixels are weighted using a quadratic function in order to generate the mosaic. Finally the *rendered image* is generated applying a similarity transform (scaling and shifting) on the *rendered image*.

sequence is used as the reference. The projection between each image and the reference image is obtained using the concatenation of successive homographies ($H_{1 \rightarrow k+1} = (\prod_{j=1}^k H_{j \rightarrow j+1})$). Finally, source images are blended using the quadratic filter $f(u, v) = (1 - (u - \frac{ImgWidth}{2})^2)(1 - (v - \frac{ImgHeight}{2})^2)$. Figure 4.6 presents a mosaic building based on a sequence of 450 images taken from a

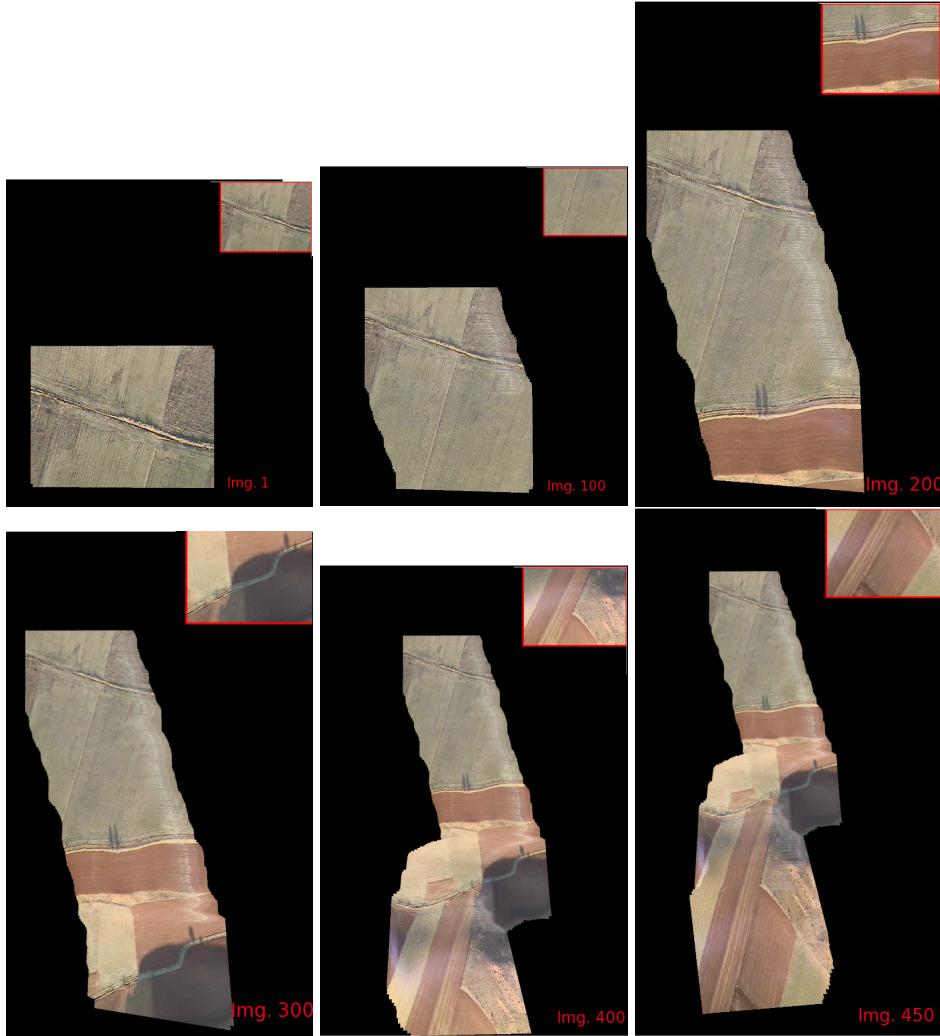


Figure 4.6: Mosaic building based on a sequence of 450 images taken from a UAV. The building process are show in images 1, 100, 200, 300, 400 and 450. Upper corner red box correspond to the current frame on the image sequence. When the projected pixel goes outside the render image boundary, a new similarity function is calculated in order to scale all the mosaic to the rendered image.

UAV flying at fixed altitude. In this sequence, the rendered image is selected to by three times the size of the reference image. As soon as the projected pixel goes outside the rendered image boundary, a new similarity transformation is generated in order to rescale the mosaic, allowing that the mosaic always are on the cover area of the rendered image.

Figure 4.7 shows two mosaics building examples. The first one is with a handheld camera, while the second one is using the Colibri 3 UAV helicopter (Appendix B.1). Videos showing the mosaics building process based on UAV

images are available on the vision4uav project web page (CVG-UPM, 2010).

4.4. UAV 3D pose estimation based on planar landmarks

This section explains the use of a pose estimation method based on frame to frame object tracking using robust homographies. The method described, makes a matching between consecutive images of a planar reference landmark, using either, the pyramidal Lucas Kanade optical flow on corners detected using the method proposed by Shi and Tomasi (Shi and Tomasi, 1994), or the ICIA algorithm (Buenaposada et al., 2003) for an object template appearance tracking. The frame to frame matching is used to estimate a projective transformation between the reference object and the image, using it to obtain the 3D pose of the object with respect to the camera coordinate system.

Considering these test, a Monochrome CCD Firewire camera with a resolution of 640x480 pixels is used. The camera is calibrated before each test, so the intrinsic parameters are known. The camera is installed in such a way that it looks downward with reference to the UAV. A rectangular helipad is used as the reference object to which estimate the UAV 3D position. It is aligned in such a way that its axes are parallel to the local plane North East axes. This helipad was designed in such a way that it produces many distinctive corners for visual tracking. Figure 4.9, shows the helipad used as reference and Figure 4.10, shows the coordinate systems involved in the pose estimation.

The algorithm begins, when a user manually selects four points on the image that corresponds to four points on a rectangle in the scene, forming the matched points $(0, 0) \leftrightarrow (x_1, y_1)$, $(910 \text{ mm}, 0) \leftrightarrow (x_2, y_2)$, $(0, 1190 \text{ mm}) \leftrightarrow (x_3, y_3)$ and $(910 \text{ mm}, 1190 \text{ mm}) \leftrightarrow (x_4, y_4)$. This manual selection generates a world plane defined in a coordinates frame in which the plane equation of Π is $Z = 0$ and also defining the scale for the 3D results. With these four correspondences between the world plane and the image plane, the minimal solution for homography \mathbf{H}_w^0 is obtained.

Once the alignment between the camera coordinate system and the reference helipad are known (\mathbf{H}_w^0) the homographies between consecutive frame are estimated, using either, the pyramidal LKT or the ICIA algorithm as is described below:

Optical flow and RANSAC: good features to track are extracted on the zone

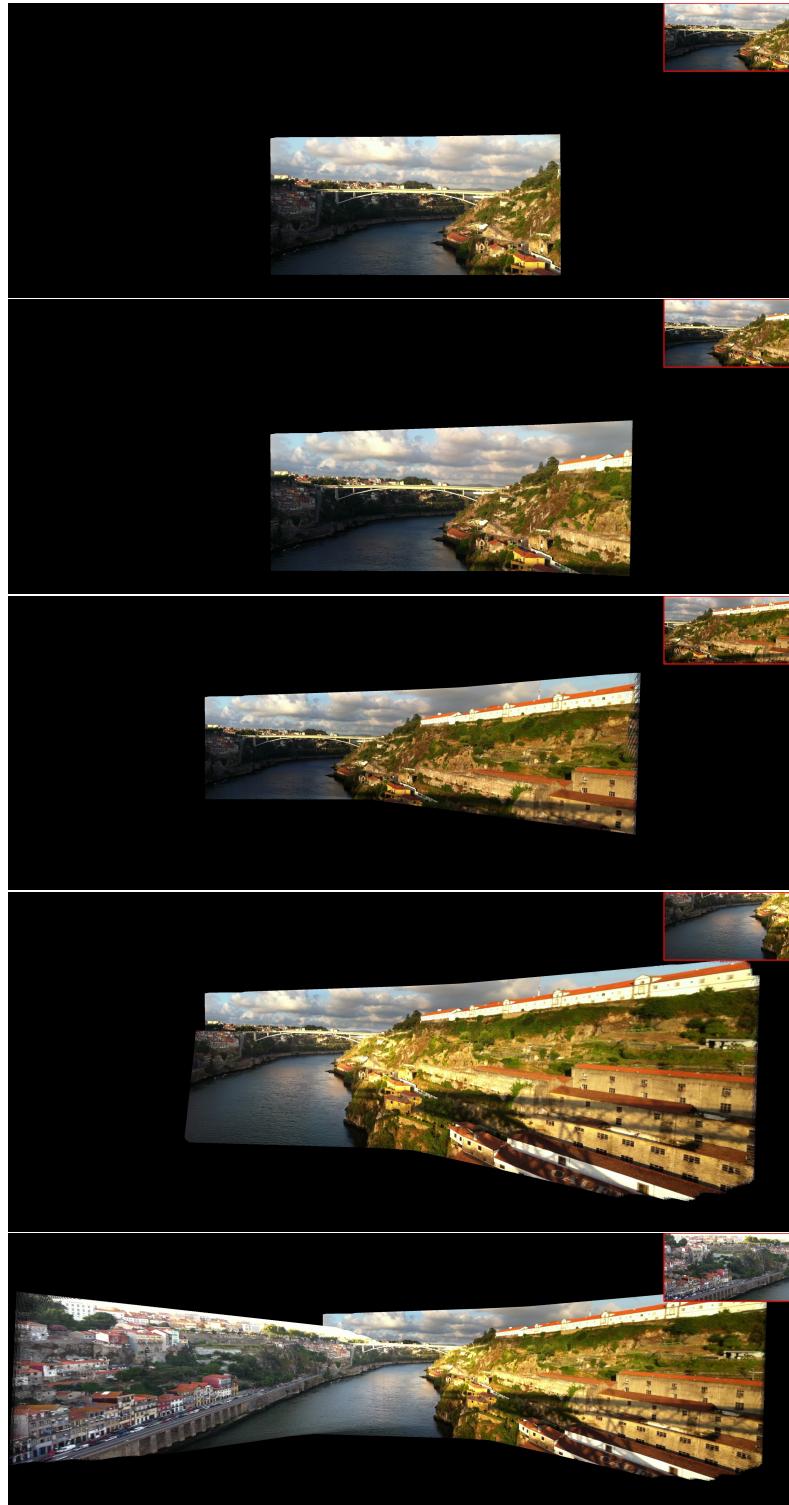


Figure 4.7: Mosaic building based on a sequence of 450 images taken with a hand held camera. The building process are show in images 50, 100, 200, 300, 500 and 620.



Figure 4.8: Mosaic Building based on 300 images taken with the Colibri 3 UAV. The building process are show in images 50, 100, 200, and 300. Bottom is a view of the Colibri 3 UAV during this mosaic building test.

corresponding to the projection of the helipad on image I_0 . Then a new image I_1 is captured, and for each one of the corners on image I_0 , the pyramidal implementation of the Lucas Kanade optical flow method is applied.

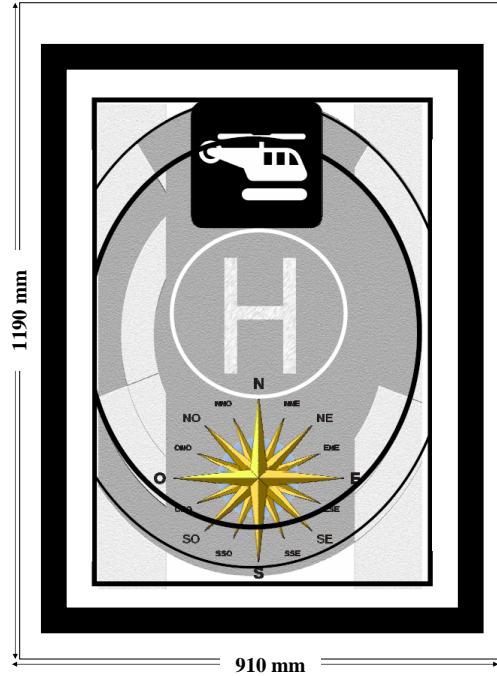


Figure 4.9: Helipad used as a plane reference for UAV 3D pose estimation based on homographies.

It obtains for each one either the corresponding position (velocity vector) on image I_1 (if the corresponding point was found on the second image) or "null" if it was not found. With matched points found on image I_1 , a homography \mathbf{H}_0^1 is robustly estimated using the algorithm described on section 3.2.3. Homography \mathbf{H}_0^1 is used to estimate the alignment between image I_1 and the reference helipad using $\mathbf{H}_w^1 = \mathbf{H}_0^1 \mathbf{H}_w^0$, which is used in order to obtain the rotation matrix \mathbf{R}_w^1 and the translation vector \mathbf{t}_w^1 using the method described on section 3.2.3. Then, the original frame formed by points $((x_1, y_1), (x_2, y_2), (x_3, y_3)$ and (x_4, y_4)) are projected onto the image I_1 using $\mathbf{x}_{I_1}^i = \mathbf{H}_0^1 \mathbf{x}_{I_0}^i$, defining the actual position of the helipad on the image I_1 . For this position, good features to track are once again estimated and used for calculating a new set of matched points between images I_1 and I_2 . These set of matched points are used to calculating \mathbf{H}_1^2 , and then \mathbf{H}_0^2 and \mathbf{H}_w^2 from which \mathbf{R}_w^2 and \mathbf{t}_w^2 is estimated. The process is successively repeated until either, the helipad is lost or the user finishes the process.

ICIA: The zone corresponding to the projection of the helipad on image I_0 is defined as the template to track $T(\mathbf{x})$ on the image sequence. Then for each new image I_k on the sequence, the following equation $\sum_{\forall \mathbf{x} \in X} (I_k(W(\mathbf{x}; \mu) - T(\mathbf{x}))^2$ is minimized in order to get the parameters $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ for a homography motion model (section 3.2.1), obtaining directly the homography \mathbf{H}_0^k that relates the image I_k with the template $T(\mathbf{x})$ on image

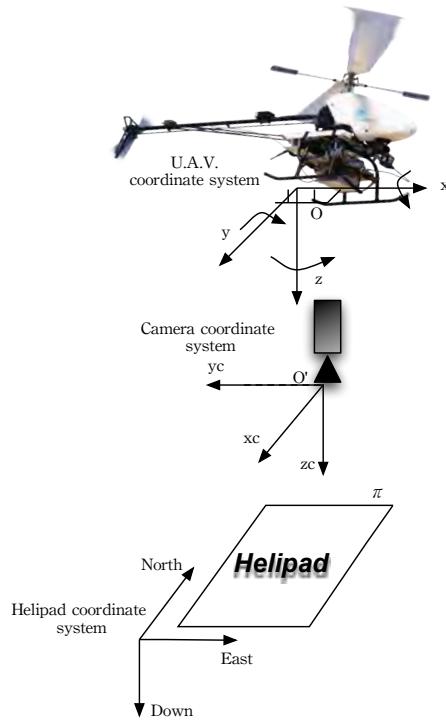


Figure 4.10: Helipad, camera and U.A.V coordinate systems.

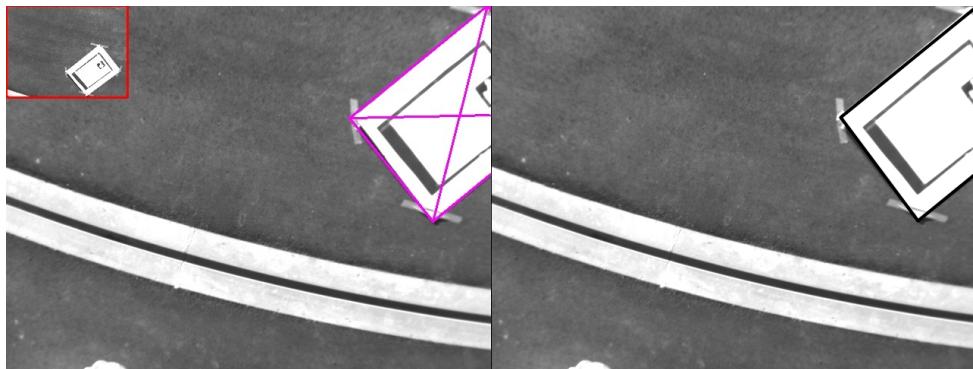


Figure 4.11: Homography motion model estimated using either, the Lucas Kanade Algorithm with RANSAC robust function fitting (left) or with the Inverse Compositional Algorithm ICIA (right). Superimposed (top left), is the original frame template under tracking.

I_0 . The alignment between frame k and the world plane is obtained using $\mathbf{H}_w^k = \mathbf{H}_w^0 \mathbf{H}_0^k$ from which \mathbf{R}_w^k and \mathbf{t}_w^k is estimated.

Figure 4.11 shows the homography estimation using both, the pyramidal Lucas Kanade tracker and the ICIA algorithm.

The translational vector obtained using the method described on section 3.3.1, is already scaled based on the dimensions defined for the reference plane during the alignment between the helipad and image I_0 , so in our case the resulting vector \mathbf{t}_w^i is in mm. The rotation matrix can be decomposed on Tait-Bryan or Cardan angles. The Tait-Bryan or Cardan angles are formed when three rotations sequences each occur about a different axis. This is the preferred sequence in flight and vehicle dynamics. Specifically, these angles are formed by the sequence: (1) ψ about z axis (*Yaw*), (2) θ about y_a (*Pitch*), and (3) ϕ about the final x_b axis (*Roll*), where a and b denote the second and third stage in a three-stage sequence or axes. This set of rotation sequences is defined by the elementary rotation matrices as Equation 4.3 shows:

$$\begin{aligned}\mathbf{R}_{z,\psi} &= \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{R}_{y,\theta} &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \\ \mathbf{R}_{x,\phi} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}\end{aligned}\tag{4.3}$$

The final coordinate transformation matrix for Tait-Bryan angles is defined by the composition of the rotations $\mathbf{R}_{x,\phi}\mathbf{R}_{y,\theta}\mathbf{R}_{z,\psi}$ forming the Equation 4.4.

$$\mathbf{R}_{Tait-Bryan} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix}\tag{4.4}$$

The angles ψ , θ and ϕ can be obtained from the rotation matrix \mathbf{R}_w^i (considering the rotation sequence order) using the Equation 4.5.

$$\mathbf{R}_{Tait-Bryan} = \mathbf{R}_0^i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

where

$$\theta = -\arcsin(r_{13}) \quad (4.5)$$

$$\psi = \arcsin\left(\frac{r_{12}}{\cos\theta}\right)$$

$$\phi = \arcsin\left(\frac{r_{23}}{\cos\theta}\right)$$

Equation 4.5 is singular when $\theta = 0$ or $\theta = \pi$.

Figure 4.12 shows some examples of the 3D pose estimation, based on a reference helipad. This Figure shows the original reference image, the current frame, the optical flow between last and current frame, the helipad coordinates in the current frame camera coordinate system and the Tait-Bryan angles obtained from the rotation matrix.

The systems have been validated on rotary wing UAVs, showing a correct pose estimation for a working range from 0.5 m up to 12 m height. Above this altitude has not been tested.

4.4.1. 3D Pose data filtering

A Kalman Filter KF has been incorporated in the 3D pose estimation algorithm in order to smooth the position and correct the estimation errors caused by the homography drift along time. The state vector is defined as the position $[x_k, y_k, z_k]$ and velocity $[\Delta x_k, \Delta y_k, \Delta z_k]$ of the k^{th} helipad expressed in the on-board camera coordinate system. The dynamic model as a linear system where system has constant velocity is considered, as defined in Equations 4.6 and 4.7:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k \quad (4.6)$$

$$\begin{bmatrix} x_k \\ y_k \\ z_k \\ \Delta x_k \\ \Delta y_k \\ \Delta z_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \\ \Delta x_{k-1} \\ \Delta y_{k-1} \\ \Delta z_{k-1} \end{bmatrix} + \mathbf{w}_{k-1} \quad (4.7)$$



Figure 4.12: Two different test for 3D pose estimation based on a helipad tracking using robust homography estimation. The reference image is on the small rectangle on the upper left corner. Left image is the current captured frame. Right image shows the optical flow between the actual and last frame. Superimposed info corresponds to the translation vector and the Tait-Bryan angles.

Where \mathbf{x}_{k-1} is the state vector (position and velocity), \mathbf{F} is the system matrix, \mathbf{w} the process noise, and Δt represents the time step.

Because the visual system only estimates the position of the helipad, the measurements are expressed as follows:

$$\mathbf{z}_k = \begin{bmatrix} \bar{x}_k \\ \bar{y}_k \\ \bar{z}_k \end{bmatrix} + \mathbf{v}_k \quad (4.8)$$

Where \mathbf{z}_k is the measurement vector and $[\bar{x}_k, \bar{y}_k, \bar{z}_k]^t$ is the position of the helipad with respect to the camera coordinate system and \mathbf{v}_k is measurement noise. With the previous definitions, the two phases of the filter Prediction and Correction can be formulated as presented in (Welch and Bishop, 1995), assuming that the process noise \mathbf{w}_k and the measurement noise \mathbf{v}_k are white, zero-mean, gaussian noise with covariance matrix \mathbf{Q} and \mathbf{R} , respectively. These matrices are experimentally obtained.

The output of the filter is the smoothed position of the helipad, that is compared with the UAV estimation as presented in next section. These values

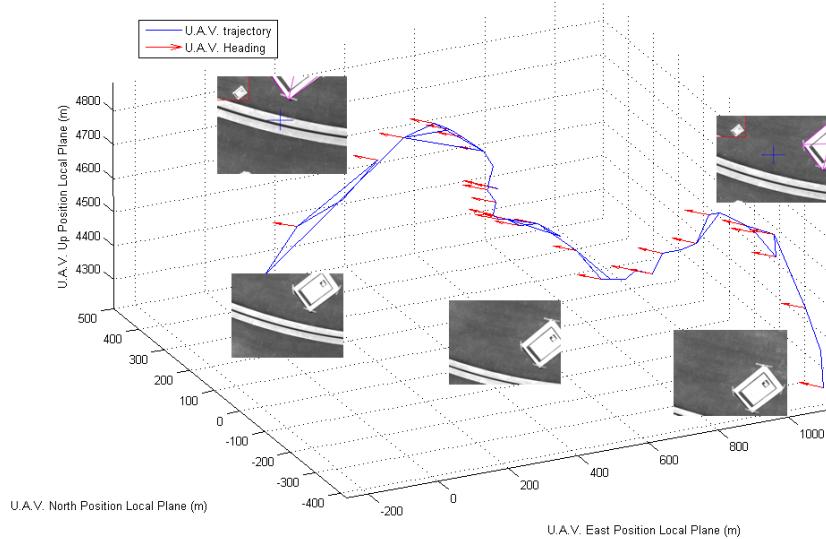


Figure 4.13: 3D flight and heading reconstruction. Superimposed images show the helipad at different times of the test, from which the 3D position is estimated.

are also used as input signals for the control system presented on chapter 5.

4.4.2. UAV 3D pose tests and results

This section shows the pose estimation tests using the Colibri 3 Electric UAV (CVG-UPM, 2010). For these tests a series of flights in autonomous mode at different heights were performed. The test begins when the UAV is hovering over the helipad, a moment in which the helipad is detected, tracked and used for estimating the 3D position of landmark w.r.t aircraft.

The tests presented in this section correspond to two different 3D pose estimation, as shown in Figure 4.12. The helicopter is positioned at two different flight levels. The first test, aircraft is hovering at 4.2 m. The second is the test that begins with a height of 10 m. A 6 mm lens on a 640x480 monochrome Firewire calibrated camera is employed. The system runs on a 1.5 GHz Via nano-ITX computer on-board with 2GB RAM with an average of 12 fps. Figure 4.13 shows the reconstruction of the flight test 1, using the IMU+GPS data.

The estimated 3D pose is compared with helicopter position estimated by the autopilot (IMU+GPS data) on the local plane with reference to the takeoff point (center of the Helipad). Because the local tangent plane to the helicopter is defined in such a way that the X axis is the North position, the Y axis is the East position and Z axis is the Down Position (negative), the Measured X and Y values must be rotated according with the helicopter heading or *Yaw*

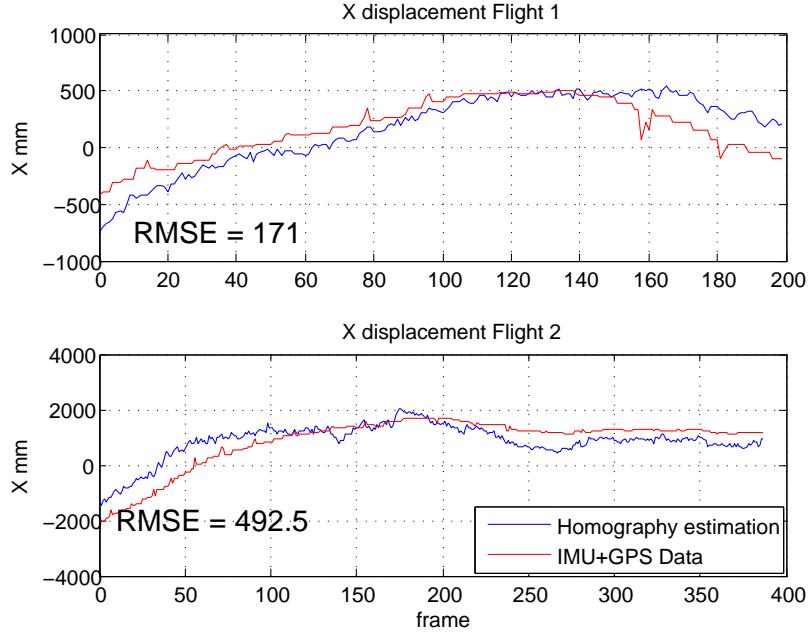


Figure 4.14: Comparison between the X axis displacement for homography estimation and IMU+GPS state estimated data. Average frame rate is 12 Hz.

angle, in order to be comparable with the estimated values obtaining from the homographies. Figures 5.10, 5.11 and 5.12 shows the landmark position with respect to the UAV and Figure 5.13, shows the estimated Y aw angle.

Results show a good performance of the visual values compared with the IMU+GPS state estimated data. In general, estimated and state estimation data have the same behavior for both test sequences. For X and Y , there is a small error between the aircraft pose state and the values estimated using the visual system, giving a maximum root mean squared error RMSE of 0.42 m in X axis and 0.16 m in Y axis. The estimated altitude position Z have a small error for flight 1 with a RMSE of 0.16 m and 0.85 m in test 2. Although results are good for height estimation, is important to remember that the state altitude estimation has an accuracy of ± 0.5 m, causing that the reference altitude estimation used to validate our approach have a big uncertainty. Finally, the Y aw angle is correctly estimated, presenting for the first flight an error of 2° between the IMU and the estimated data, and 4° for the second tests.

Results have also shown that the system correctly estimate the 3D position when a maximum of the 70 % of the landmark is partially occluded or out of the camera field of view as Figure 4.18 shows. There is an available video showing

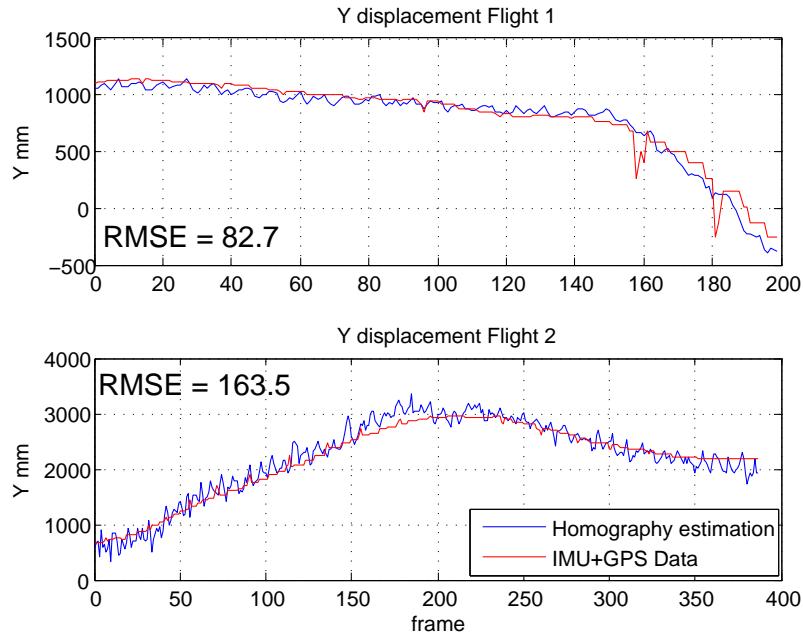


Figure 4.15: Comparison between the Y axis displacement for homography estimation and IMU+GPS state estimated data. Average frame rate is 12 Hz.

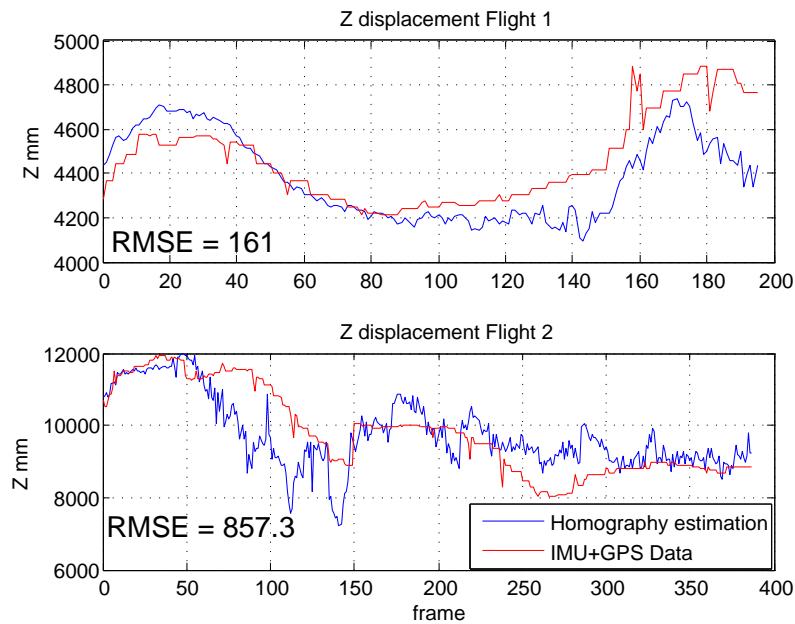


Figure 4.16: Comparison between the Z axis displacement for homography estimation and IMU+GPS state estimated data. Average frame rate is 12 Hz.

the sequences of the test explained in this section, available at the vision4uav project web page (CVG-UPM, 2010).

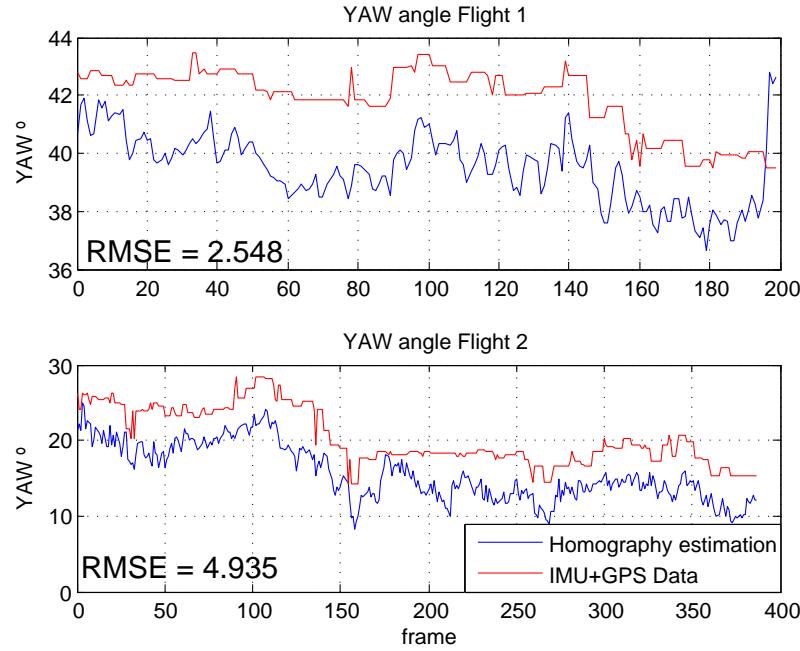


Figure 4.17: Comparison between the *Yaw* angle measured using homography estimation and IMU+GPS state estimated data. Average frame rate is 12 Hz.

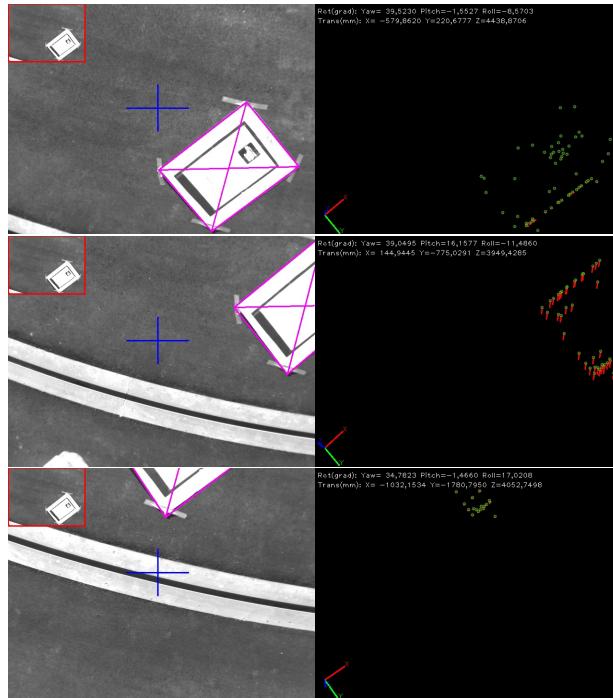


Figure 4.18: 3D pose estimation occlusion robustness. The system correctly estimate the 3D position when a maximum of the 70 % of the landmark is partially occluded or out of the camera field of view.

4.5. Conclusions

This chapter has shown how to exploit on-board visual data on UAVs. The proposed methods are based on visual tracking and robust projective transformation estimation, showing the versatility for tasks like video stabilization, mosaic building and 3D pose estimation among others.

This chapter has introduced a robust 3D pose estimation system for UAVs based on piecewise planar object tracking using homographies. The method was validated on real UAV flights, and results have shown that estimated data is comparable in precision and quality with the one obtained by the IMU+GPS state estimated of the on-board controller. Results have shown that the 3D pose estimated with a frame rate of 12 fps by the visual system is consistent with the position calculated by the on-board controller. Tests have been done at different altitudes, showing that the system correctly works in a range of 0.5 up to 12 m.

The quality of the object tracking system by using a robust frame-to-frame homography estimator has also been tested. The object can be correctly tracked and its 3D position obtained with high precision, when at least 30 % of the reference object is not occluded or out of the camera field of view, as video sequences in the results shows.

Visual processing techniques have also been used for developing two real time video enhancement systems for UAVs. The first technique is a video stabilization system for light UAVs without sophisticated camera platforms or gimbals. It reduces the effect caused by vibrations and sudden aircraft movements, and is robust against large image movements and camera focal distance changes. The second technique, allows building real time aerial mosaics that are useful on aerial photogrammetry and map building process.

Chapter 5

Visual Control for rotary wing UAV

5.1. Introduction

Stability control issue of VTOL UAVs has been well studied, and can be considered as solved. Nevertheless, it is not the case of visual control techniques that improve robot capabilities. High robustness, performance and reliability are generally required in high level controls. It thereby requires new developments on computer vision techniques needed to fulfill these specifications. In addition, it correctly must drives common issues presented on aerial robotics, like a dynamic working space with external perturbations and the non-holonomic aircraft behavior, among others.

In this chapter, two control strategies for UAVs are presented. The first strategy uses the 3D camera (aircraft) pose estimation method presented in chapter 4 to develop a decoupled servoing system suitable for landing and hovering positioning tasks on VTOL aircrafts. The second uses visual references on the image plane in order to directly command a rotary wing UAV to continuously follow an aerial moving target in a task known as "object following". Both controllers use the principles of visual servoing theory, adapting them to drive the non-holonomic behavior of rotary wing UAVs.

The chapter, firstly makes a review of visual servoing techniques, and then

shows how they have been adapted by taking into account flight dynamics and the visual task in order to create decoupled controllers. It allows having almost pure and direct visual control systems. Both controllers are validated on real test using rotary wing UAVs.

5.2. Visual Servoing

Visual servoing is defined as the process of using computer vision information for controlling the movement of a robot, i.e., the used of visual data in the control loop (Chaumette and Hutchinson, 2006), (Hutchinson et al., 1996). Computer vision is a non intrusive sensor capable of giving fast and accurate information of a working space. This information is used for modern control systems in order to make accurate control actions on mobile robots and manipulators.

In the literature, there are different ways in defining and classifying a visual servoing system. They can be classified according to the number of cameras and the distribution with respect to the task space as is explained on section 5.2.1. There are also different taxonomies used to define a visual control system, such as the one proposed by Sanderson and Weiss (Sanderson and Weiss, 1983) which is most widely used. It divides the control structures according to the way the visual system interacts with the overall system and is briefly explained in section 5.2.2. A second common used taxonomy considers the way how visual information is processed and used for defining the control systems as is presented in section 5.2.3. The explained taxonomies allows us to introduce the developed contributions on the control of UAVs based on visual information obtained from different algorithms used to estimate the 3D pose and attitude of the aircraft explained on previous chapters.

5.2.1. Camera Configurations

Visual Servoing applications traditionally have been built using two camera configurations. The first configuration has a camera mounted on the robot end-effector and is often known as *eye-in-hand* configuration. In the second called *eye-to-hand* configuration, there is at least one fixed camera that has a global view of the scene, including the end-effector and the workspace. Both configurations are explained below:

Eye-in-hand Configuration

In this configuration, the camera unison moves with an end-effector that allows having a fixed and known relationship between the pose of the end-effector and the pose of the camera. The principal characteristic of this configuration is

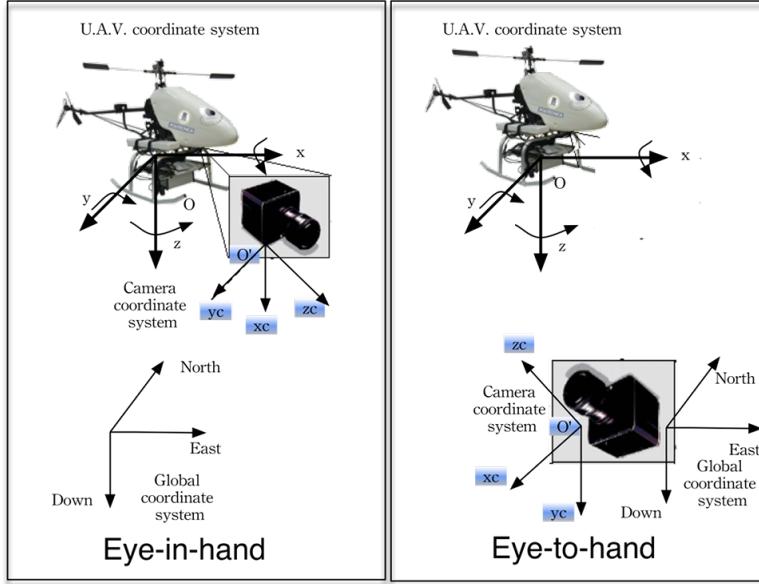


Figure 5.1: Camera configuration for UAV visual servoing. Left: Eye-in-hand configuration. Right: Eye-to-hand configuration. Combinations of both are also possible.

that the field of view FOV of the camera is dynamic and allows having a changing scene. It is possible to control the actuator in such a way that it allows it to obtain higher details of object of interest and the general scene. However, occlusions can also happen caused by other objects that appear on the camera FOV, and thereby losing the target by end-effector movements. Figure 5.1 shows this camera configuration in a UAS.

Eye-to-hand Configuration

In this configuration, the camera is fixed on some point of the workspace in such a way that the camera observes the scene, the end-effector and the objects of interest. The relationship between the camera and the base coordinate system of the mobile robot is known. However, the pose of the camera is independent of the end-effector movement, making it necessary to estimate the pose relation between the camera, the end-effector and its base coordinate system during the controls task. In this configuration there is a global view of the actuator and the object of interest in the scene. However, it has less details than in the case of the eye-in hand configuration. Figure 5.1 show this camera configuration in a UAS.

Hybrid Configuration

It is a combination of the eye-in-hand and eye-to-hand configuration that tries to get the best of both system, the detailed local information provide by the camera mounted on the end-effector and the global view of the scene provide by the fixed camera. This configuration is also designed in order to reduce possible occlusion caused by the end-effector and other objects on the field of view of both cameras. In this category can also be included systems in which there are cameras mounted on diverse actuators, as with in collaborative mobile robots.

5.2.2. Visual control schemes

Traditionally, there have been several approaches to visual control schemes. One of the first proposed taxonomies was based, in a way that the visual information interacts with the overall system (Sanderson and Weiss, 1983). Posterior taxonomies are based on the selection of the visual features and the way that the control error function is defined (Weiss, 1984). There are also others taxonomies in which having a 3D scene model or the necessity of having some camera calibration parameters is obligatory.

Following, there is brief description of some of the most common taxonomies, specially these that are focused on velocity controls, which are generally common, appropriated and generalized control strategies in a variety of mobile robots and manipulators, as is indicated in (Chaumette and Hutchinson, 2006).

Visual control schemes according to the system interaction

The principal criteria for the definition of this taxonomy, is according to the way the visual controller interacts with the robot actuators, allows having two main classes (Hutchinson et al., 1996). The first configuration, visual control systems, which generate control signals that are directly sent to the robot end-effector and designated as *direct visual servoing*. The second, visual control that generates a signal used as an input reference for a low level controller, i.e., a hierarchical architecture known as *dynamic look-and-move*.

Direct Visual Servoing

In this configuration, there is a unique control loop that involves visual systems and the robot end-effectors. The visual system generates a reference signal that is directly transmitted to the actuators, making the visual system necessary to run with a frequency greater than, or at least equal to the frequency needed for the actuators to correct operation. It implies that control loop must be closed at

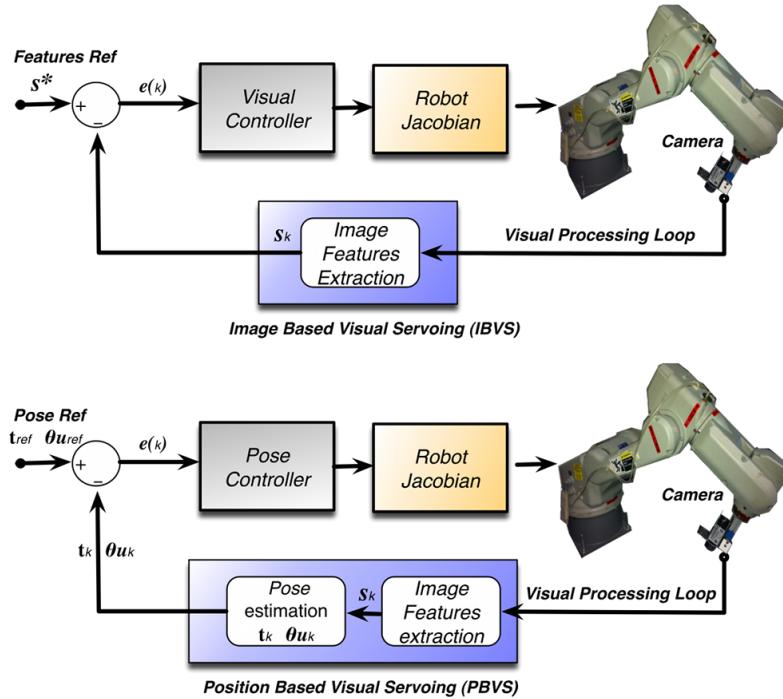


Figure 5.2: Direct Visual Servoing. Top: An Image-based visual servo IBVS. Bottom: A Pose-based visual servo PBVS.

higher frequencies (e.g., less than 10 ms). The visual controller is also responsible for maintaining the system stable. Figure 5.2 shows a Direct visual servoing based on either image error or position error.

Dynamic look and move

This scheme is based on a hierarchical architecture, the vision system provides set points used as a reference for a low level controller. This internal control loop stabilizes the robot and generates actuator actions in order to achieve the desired posed according to the input commands and the system dynamics. This is one of the most extended architectures, caused in part by the relative low sampling rates necessary to command the internal control loop and the capacity of modern robots and manipulators to accept Cartesian velocities or incremental positions commands. The higher frequency of the low level control loop (compared with the visual system) that allows idealizing the dynamics of the robot (i.e., it is an ideal Cartesian motion device), separating from the visual controller the dynamics and kinematic singularities of the mechanical system, and simplifying the construction of the visual controller, making it more portable (Hutchinson et al., 1996). Figure 5.3 shows a Dynamic look-and-move control based on either image error or position error.

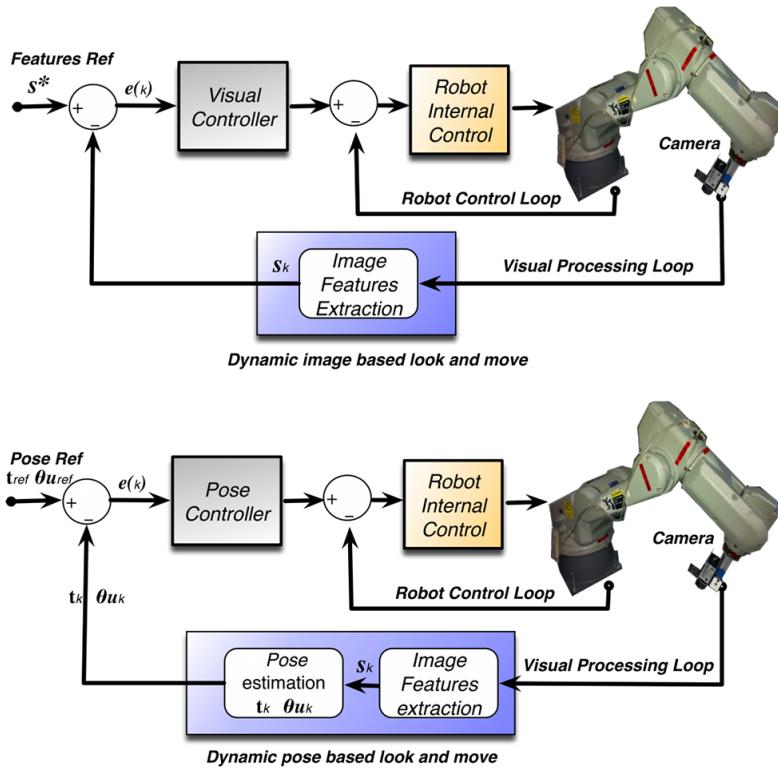


Figure 5.3: Dynamic look and move control. Top: An Image-based look-and-move control. Bottom: A Pose-based look-and-move control.

5.2.3. Visual control schemes according to the definition of the error function

There are two classical approaches for visual servoing schemes that are defined according to the workspace in which the control is developed. The first is *image-based visual servoing* IBVS, in which the workspace is enclosed directly on the image field. This control scheme uses the image plane coordinates as a set of points in order to define the error function that the control system minimizes. The second classical approach known as *position-based visual servoing* PBVS, where the workspace consists of a set of three dimensional parameters that are estimated from the image space. This control uses the pose of the camera with respect to some reference frame in order to define the function error. The camera pose is computed from a set of features on the image plane and additional information like the camera calibration and some knowledge of the scene. These two methods are explained with some detail below. There are also advanced control schemes, like the hybrid control method that fuses 3D information obtained from the PBVS and the 2D information obtained from the image space in order to use the advantages of both methods (Malis et al., 1999). This control scheme is focused on decoupling the rotations motion from the translational, using 3D information to control the translational error and

2D information to control the rotational error. In a similar way to the hybrid method, the partitioned visual scheme tries to make a full decoupling of each of the translational and rotational motions, by means of finding six features such as each is related with a unique degree of freedom, generating a diagonal Jacobian or interaction matrix whose elements are constant, leading to a simple linear control. A detail explanation of these advanced approaches can be found in (Malis et al., 1999), (Siciliano and Khatib, 2008).

Image-based visual servoing

In this control scheme, error function and control law are defined from characteristics taken directly from the image plane. This control scheme reduces the amount of processed information because estimating the position of the object of interest is not needed such as in the case of the position control scheme.

The classical IBVS uses the image features to define a characteristic \mathbf{s} , from which the controller is designed. The features can be defined either directly on pixels ($\mathbf{s} = (u, v)$) or in central coordinates of the image plane ($\mathbf{s} = (x, y)$). For this scheme, it is not necessary to know the camera intrinsic parameter or calibration matrix \mathbf{K}_c . As proposed in (Chaumette and Hutchinson, 2006), the error function can be defined as the difference between the current characteristics (measured on the image) and its desired position on the image ($\mathbf{e} = \mathbf{s} - \mathbf{s}^*$).

Although the error \mathbf{e} function is defined on the image plane, the control must generate adequate reference for the end-effector low level controller. These references often are defined in a cartesian coordinate system. Therefore, in order to build a servo controller, it is necessary to relate changes in the image space to changes in the position of the end-effector. This relation is usually defined using the Jacobian (also called the interaction matrix).

One of the most common and straightforward approaches for visual servoing is the design of a velocity controller (i.e., references to the low level control are instantaneous velocity commands). In order to obtain the Jacobian, it is necessary to obtain the relationship between the time variation of \mathbf{s} and the camera velocity. Assuming that there is a point in the space with coordinates on the camera frame $\mathbf{X}^c = [X, Y, Z]^T$, it is projected with normalized coordinates on the image plane as $\mathbf{s} = (x_n, y_n)$ throughout a pinhole camera projection model as equation 5.1 shows:

$$\begin{aligned} x_n &= \frac{X}{Z} = \frac{u - c_u}{f\alpha} \\ y_n &= \frac{Y}{Z} = \frac{v - c_v}{f} \end{aligned} \tag{5.1}$$

where the parameters (c_u, c_v, f, α) are the set of camera intrinsic parameters.

The time derivative of the projected point on the image plane is defined in Equation 5.2:

$$\begin{aligned} \dot{x}_n &= \frac{\dot{X}}{Z} - X \frac{\dot{Z}}{Z^2} \\ \dot{y}_n &= \frac{\dot{Y}}{Z} - Y \frac{\dot{Z}}{Z^2} \end{aligned} \quad (5.2)$$

In order to relate the velocity of the point \mathbf{X}^c to the camera spatial velocity, lets define the spacial velocity of the camera as $\mathbf{V}_c = (\mathbf{v}_c, \boldsymbol{\omega}_c)$, with $\mathbf{v}_c = [v_x, v_y, v_z]^T$ the camera's instantaneous linear velocity and $\boldsymbol{\omega}_c = [\omega_x, \omega_y, \omega_z]^T$ the instantaneous angular velocity of the camera. The time derivative of the point \mathbf{X}^c is defined as $\dot{\mathbf{X}} = -\mathbf{v}_c - \boldsymbol{\omega}_c \times \mathbf{X}$ as shown in Equation 5.3.

$$\begin{aligned} \dot{X} &= -v_x - \omega_y Z + \omega_z Y \\ \dot{Y} &= -v_y - \omega_z X + \omega_x Z \\ \dot{Z} &= -v_z - \omega_x Y + \omega_y X \end{aligned} \quad (5.3)$$

Integrating equation 5.3 in Equation 5.2 and considering the projection model defined by equation 5.1 we obtain equation 5.4.

$$\begin{aligned} \dot{x}_n &= -\frac{v_x}{Z} + \frac{x_n v_z}{Z} x_n y_n \omega_x + (x_n^2 + 1) \omega_y + y_n \omega_z \\ \dot{y}_n &= -\frac{v_y}{Z} + \frac{y_n v_z}{Z} + (y_n^2 + 1) \omega_x - x_n y_n \omega_y - x_n \omega_z \end{aligned} \quad (5.4)$$

Equation 5.4 can be written as $\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{V}_c$, where \mathbf{L}_s is the Jacobian or interaction matrix as shown in Equation 5.5.

$$\mathbf{L}_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x_n}{Z} & x_n y_n & -(x_n^2 + 1) & y_n \\ 0 & -\frac{1}{Z} & \frac{y_n}{Z} & (y_n^2 + 1) & -x_n y_n & -x_n \end{bmatrix} \quad (5.5)$$

Equation 5.5 is a function of depth (Z) of the point \mathbf{X}^c relative to the camera frame. Therefore, a controller designed using this Jacobian, must estimate this value. Because there is not alway possible to know the value of Z , usually it is used a variation of the interaction matrix also known as the Estimated Jacobian $\widehat{\mathbf{L}}_s$. There are several ways to calculate the Estimated Jacobian. One popular scheme is trying to estimate the value of Z as real as it can, using an adaptive method on each iteration (Hutchinson et al., 1996). Another popular method, defines Z as a constant value being equal to the desired final pose $Z = Z^*$ ($\mathbf{e} = 0$). In

this case $\widehat{\mathbf{L}}_s$ is constant, and only the desired depth of each point must be defined.

If \mathbf{V}_c is selected as the desired low level control input, the relationship between the camera velocity and the time variation of the error can be defined as $\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{V}_c$, where $\mathbf{L}_e = \mathbf{L}_s$. The error function is often selected in such a way that there is an exponential decoupled decrease in error ($\dot{\mathbf{e}} + \lambda \mathbf{e} = 0; \lambda > 0$), allowing that the velocity commands transmitted to the low level controller can be defined as equation 5.6 shows:

$$\begin{aligned}\mathbf{V}_c &= -\lambda \mathbf{L}_e^+ \mathbf{e} \\ \text{or when only the Estimated Jacobian is available:} \\ \mathbf{V}_c &= -\lambda \widehat{\mathbf{L}}_e^+ \mathbf{e}\end{aligned}\tag{5.6}$$

where \mathbf{L}_e^+ is the Moore-Penrose pseudo-inverse of the Interaction matrix \mathbf{L}_e , $(\mathbf{L}_e^+ = (\mathbf{L}_e^T \mathbf{L}_e)^{-1} \mathbf{L}_e^T)$, and $\widehat{\mathbf{L}}_e^+$ is the pseudo-inverse of the Estimated Jacobian, when the depth Z is not available.

The obtained controller supposes a negligible end-effector dynamics, assuming that the control input command will be perfectly accomplished, moving the camera to a desired velocity.

Finally, in order to control the six degrees of freedom, it is necessary to have a minimum of 3 features or points on the image. If there are n -features available $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n]^T$, the interaction matrix will be a stacking of the Jacobian for each of the n -features as shown in Equation 5.7.

$$\mathbf{L}_s = \begin{bmatrix} -\frac{1}{Z_1} & 0 & \frac{x_1}{Z_1} & x_1 y_1 & -(x_1^2 + 1) & y_1 \\ 0 & -\frac{1}{Z_1} & \frac{y_1}{Z_1} & (y_1^2 + 1) & -x_1 y_1 & -x_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\frac{1}{Z_n} & 0 & \frac{x_n}{Z_n} & x_n y_n & (x_n^2 + 1) & y_n \\ 0 & -\frac{1}{Z_n} & \frac{y_n}{Z_n} & (y_n^2 + 1) & -x_n y_n & -x_n \end{bmatrix}\tag{5.7}$$

Position-based visual servoing

In this control scheme, features are extracted from the image and are then used in conjunction with camera intrinsic parameters and the scene model to estimate the pose of the camera with respect to some reference coordinate system. The error functions is defined as the difference between the current and the desired pose in this coordinate workspace. Position base visual servoing

separates the computation of the control input signal from the position estimation problem

A PBVS often defines the features \mathbf{s} in terms of cartesian coordinate system to represent the camera pose and the error function. They are usually defined as a reference pose $\mathbf{s}^* = (\mathbf{t}^*, \theta\mathbf{u}^*)$ and a measured pose $\mathbf{s} = (\mathbf{t}, \theta\mathbf{u})$, in which \mathbf{t} is the translation vector and $\theta\mathbf{u}$ is the orientation, represented as rotation angle θ around the unitary vector \mathbf{u} .

The design of the controller depends on the way that the reference and measured poses are selected (features \mathbf{s} and \mathbf{s}^*). Chaummete *et. al* (Chaumette and Hutchinson, 2006), proposes two ways for defining this characteristics, that are selected considering if the reference frame is attached to the current camera frame \mathcal{F}_c , the desired camera frame \mathcal{F}_{c^*} , or a reference frame \mathcal{F}_o fixed to the object. In the first one, the translation \mathbf{t} is defined relative to the object frame \mathcal{F}_o , obtaining that the measured feature is $\mathbf{s} = (^c\mathbf{t}_o, \theta\mathbf{u})$, and the reference is $\mathbf{s}^* = (^*\mathbf{t}, \mathbf{0}_{3\times 1})$, where $\mathbf{0}_{3\times 1}$ is a zero vector. The error function can be defined as $\mathbf{e} = (^c\mathbf{t}_o - ^*\mathbf{t}_o, \theta\mathbf{u})$, an is related with the camera velocity $\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{V}_c$ through an interaction matrix defined in Equation 5.8:

$$\mathbf{L}_e = \begin{bmatrix} -\mathbf{I}_{3\times 3} & [^c\mathbf{t}_o]_\times \\ \mathbf{0}_{3\times 3} & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix} \quad (5.8)$$

where $\mathbf{I}_{3\times 3}$ is and 3×3 identity matrix, and $\mathbf{L}_{\theta\mathbf{u}}$ is the part of the Jacobian corresponding to the object orientation and it is defined by equation 5.9

$$\mathbf{L}_{\theta\mathbf{u}} = \mathbf{I}_{3\times 3} - \frac{\theta}{2} [\mathbf{u}]_\times + \left(1 - \frac{\text{sinc}\theta}{\text{sinc}^2 \frac{\theta}{2}}\right) [\mathbf{u}]_\times^2$$

where

(5.9)

$$[\mathbf{u}]_\times^2 = \text{skew } \mathbf{u} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$$

and $\text{sinc}x$ is the sinus cardinal defines such that $\text{sinc}(\theta) = \sin(\theta)/\theta$ and $\text{sinc}(0) = 1$.

To obtain the control input $\mathbf{V}_c = -\lambda \mathbf{L}_e^{-1} \dot{\mathbf{e}}$, it is necessary to obtain the inverse of the estimated Jacobian. For a six degrees of freedom DOF systems, it is defined as equation 5.10 shows.

$$\widehat{\mathbf{L}_e^{-1}} = \begin{bmatrix} -\mathbf{I}_{3\times 3} & [^c\mathbf{t}_o]_\times \mathbf{L}_{\theta\mathbf{u}}^{-1} \\ \mathbf{0}_{3\times 3} & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix} \quad (5.10)$$

Using equation 5.10 and considering that $\mathbf{L}_{\theta\mathbf{u}}^{-1}\theta\mathbf{u} = \theta\mathbf{u}$, it is possible to obtain the low level controller input, composed by the linear $\mathbf{v}_c = [v_x, v_y, v_z]^T$ and angular $\boldsymbol{\omega}_c = [\omega_x, \omega_y, \omega_z]^T$ velocities, as shown in Equation 5.11.

$$\begin{aligned}\mathbf{v}_c &= -\lambda \left(\left({}^c\mathbf{t}_o - {}^c\mathbf{t}_o \right) + [{}^c\mathbf{t}_o]_{\times} \theta\mathbf{u} \right) \\ \boldsymbol{\omega}_c &= -\lambda \theta\mathbf{u}\end{aligned}\quad (5.11)$$

Another common PBVS scheme, defines the measured characteristic as $\mathbf{s} = ({}^c\mathbf{t}_c, \theta\mathbf{u})$ and the desired as $\mathbf{s}^* = (\mathbf{0}_{3 \times 1}, \mathbf{0}_{3 \times 1})$, given that error function is $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$. It allows that the Jacobian that relates the error velocity with the camera velocity ($\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{V}_c$) presents a decoupling behavior between the translational and rotational motions as shown in Equation 5.12.

$$\mathbf{L}_e = \begin{bmatrix} {}^c\mathbf{R}_c & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix} \quad (5.12)$$

where ${}^c\mathbf{R}_c$ is the rotation matrix that gives the orientation of the current frame relative to the desired frame.

Finally, the low level controller input is defined in Equation 5.13:

$$\begin{aligned}\mathbf{v}_c &= -\lambda {}^c\mathbf{R}_c^T \mathbf{t}_c \\ \boldsymbol{\omega}_c &= -\lambda \theta\mathbf{u}\end{aligned}\quad (5.13)$$

5.3. Landing and Hovering based on visual pose control

Precision hovering and landing have been an actively studying field on recent years. Different works have been done where a vision system was used for low altitude position estimation and autonomous landing. In (De Wagter and Mulder, 2005), the authors have evaluated the use of visual information at different stages of a UAV control system, including a visual controller and a pose estimation for autonomous landing using a chessboard pattern. Saripalli *et. al.* proposed and experimental method for autonomous landing on a moving target, (Saripalli and Sukhatme, 2007). It is used for tracking a known helipad and using it to complement the controller GPS+IMU state estimation. Merz (Merz et al., 2006), uses a method similar to the one proposed here. It uses pose estimation based on homographies and a known helipad, as is explained below.

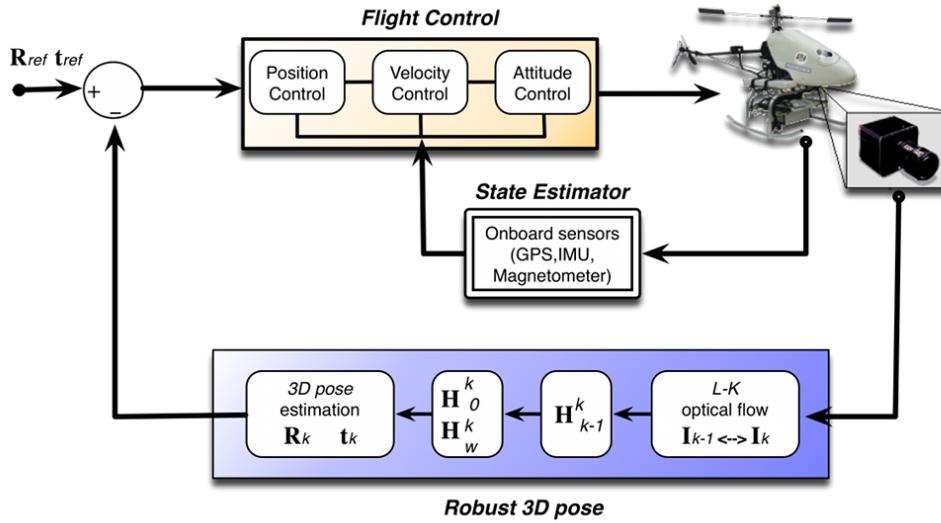


Figure 5.4: UAV control *Dynamic pose-based look-and-move* system architecture.

This section presents the integration of visual reference in order to make a visual control system for positioning tasks and autonomous landing based on visual references. The visual servoing is made up of two parts. The first is the pose estimation techniques presented in chapter 4, which are used in order to develop a visual control loop using a *Position Based Visual Servoing* architecture in a *Dynamic Look and Move configuration* ((Hutchinson et al., 1996), (Chaumette and Hutchinson, 2006), (Siciliano and Khatib, 2008)). The second, the current (\mathbf{s}) and the desired position (\mathbf{s}^*) of the UAV are used for calculating low level controller inputs (on-board flight controller), that is in charge of moving the UAV to the desired position as is shown in Figure 5.4.

3D pose estimation based on homographies is well described in previous chapters. However the most important concepts can be summarized in equations 5.14, and 5.15. The first relates the first two columns of the rotation matrix and the translation vector with the projection model given by a calibrated pinhole camera and a planar homography (Figure 3.3).

$$\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} = \lambda \mathbf{K}^{-1} \mathbf{H}_w^i = \lambda \mathbf{K}^{-1} \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix} \quad (5.14)$$

where

$$\mathbf{r}_1 = \lambda \mathbf{K}^{-1} \mathbf{h}_1, \quad \mathbf{r}_2 = \lambda \mathbf{K}^{-1} \mathbf{h}_2, \quad \mathbf{t} = \lambda \mathbf{K}^{-1} \mathbf{h}_3, \quad \lambda = \frac{1}{\|\mathbf{K}^{-1} \mathbf{h}_1\|}$$

The second obtains the third column of rotation matrix, warranting the orthonormality condition, that is then used to obtain the attitude camera angles.

$$\begin{aligned}
\mathbf{R} &= [\mathbf{r}_1 \quad \mathbf{r}_2 \quad (\mathbf{r}_1 \times \mathbf{r}_2)] = \mathbf{U}\mathbf{S}\mathbf{V}^T \\
\mathbf{S} &= \text{diag}(\sigma_1, \sigma_2, \sigma_3) \\
\mathbf{R}' &= \mathbf{U}\mathbf{V}^T \\
\theta &= -\arcsin(\mathbf{R}'_{13}) \\
\psi &= \arcsin\left(\frac{\mathbf{R}'_{12}}{\cos \theta}\right) \\
\phi &= \arcsin\left(\frac{\mathbf{R}'_{23}}{\cos \theta}\right)
\end{aligned} \tag{5.15}$$

The translational vector $\mathbf{t} = \lambda \mathbf{K}^{-1} \mathbf{h}_3$ is filtered using the process presented on section 4.4.1. Because the camera is looking downward, the obtained values are rotated to the aircraft frame ($\mathbf{t}_{UAV} = \mathbf{R}_{c2UAV} \mathbf{t}$; $\mathbf{R}_{UAV} = \mathbf{R}_{c2UAV} \mathbf{R}'$), in order to be compared with the aircraft pose references as is show in Figure 5.5. The rotated translation vector \mathbf{t}_{UAV} and the rotated *Yaw* estimation are then used to generate the low level control input using the Equation 5.13. Because the decoupled behavior of the control law and the fact that *Roll* and *Pitch* UAV angles are directly controlled by the low level autopilot, only velocity displacements and *Yaw* rotation commands are generated from the visual control system.

5.3.1. Tests and results

Positioning control

The 3D pose control system has been implemented on-board the Colibri 3 UAV (appendix B). In this scheme, the position references generated by 3D pose estimator (running on-board aircraft on a VIA nano-ITX 1.5 GHz CPU) are used as input references for the SR20 low level controller. This low level controller allows getting input velocity commands, as well as direct relative pose references. For this test, the filtered pose data generated by the visual system are directly sent as relative pose references (to be executed at a defined velocity) for the low level Autopilot. The control architecture is presented on in Figure 5.4.

For these tests a 6 mm lens and a monochrome CCD Firewire camera with a resolution of 640x480 pixels is used. The camera is calibrated before each test, so the intrinsic parameters are known. The camera is installed in such a way that it looks downward with relation to the UAV. A rectangular helipad (Figure 4.9) is used as the reference object to which estimate the UAV 3D position. It is aligned in such a way that its axes are parallel to the local plane North East

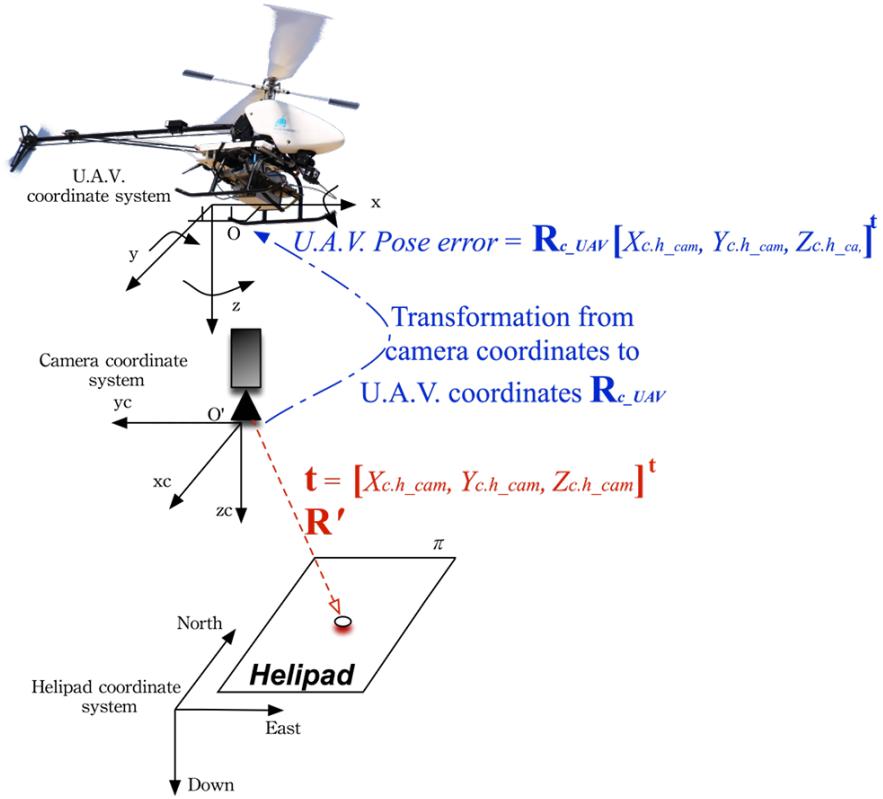


Figure 5.5: Using a helipad as a plane reference for UAV 3D pose estimation based on homographies.

axes. The UAV local plane is set to be on the center of the helipad (takeoff and landing point).

The on-board system has been used to control the UAV position, using the helipad as a reference. For this test, a defined position above the helipad (0 m, 0 m, 3 m) has been used in order to test the control algorithm. Test begins with the helicopter hovering on the helipad with an altitude of 6 m. The on-board visual system is used for estimating the relative position of the helipad with respect to the camera system; the obtained translation vector is used for sending the reference commands to the UAV's controller with an average frame rate of 12 fps. The algorithm firstly centers the helicopter on the X and Y axes, and when the position error in these axes is small to a defined threshold (0.5 m), it begins sending references to the Z axis. In Figure 5.6, the 3D reconstruction of the flight is presented.

Figure 5.7 shows the results of the positioning task. The red lines represent the position estimated by the visual system, which is used to generate relative

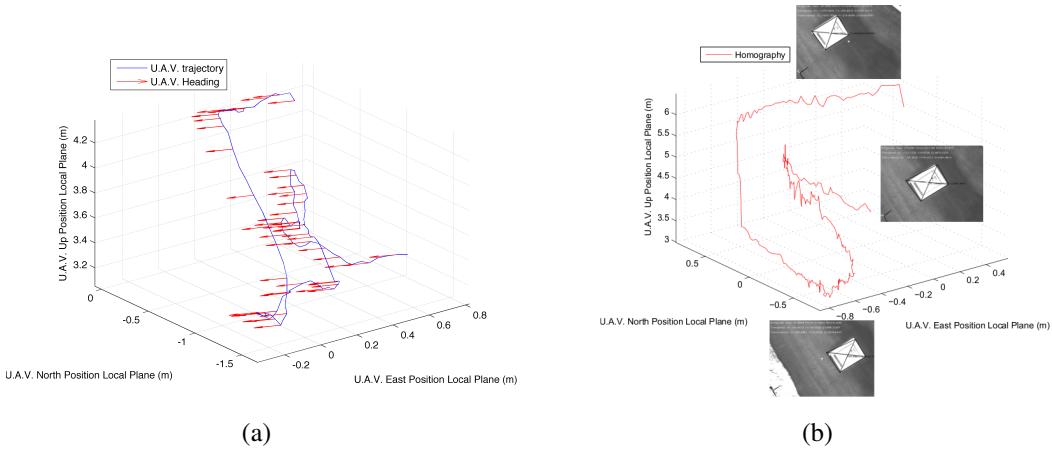


Figure 5.6: 3D reconstruction of the flight test using the IMU+GPS values (a) and the vision-based estimation (b). The blue line is the reconstructed trajectory of the UAV, and the red arrows correspond to the heading of the UAV.

position commands to the UAV controller, and the blue line represents the desired position. The test shows that the helicopter centers on X and Y axes, taking into account the resolution of the estimated pose done by autopilot using the IMU+GPS data. However, the altitude error is little higher because the Z axis precision is above 1 meter. Therefore, the IMU+GPS estimated pose precision causes that small references sometimes are not executed because they are smaller than the UAV pose resolution.

Landing control

Following a similar approach to the one presented above, the information provided by the visual system is employed by a fuzzy controller (Olivares-Méndez et al., 2010) in order to control the altitude of a rotary wing UAV, that has been proved to be useful for landing on fix flat surfaces.

For these tests, a reference point (0,0,0 m) is defined w.r.t. the helipad. Figure 5.8 shows the first image frame for each one of three different tests. Considering limited field of view of the camera and the fact that it is looking downward, the helipad is not correctly tracked below a height of 0.5 m. In order to surpass this issue, when the visual system estimates an altitude below a threshold of 0.7 m, the helicopter is commanded to descend, maintaining a constant velocity of 0.2 m/sec on Z axis. The control in X and Y axes is affected by the ground effect below this threshold limit. However, the 0.2 m/sec descending speed is defined in order to reduce this effect as recommended by the UAV manufacturer.

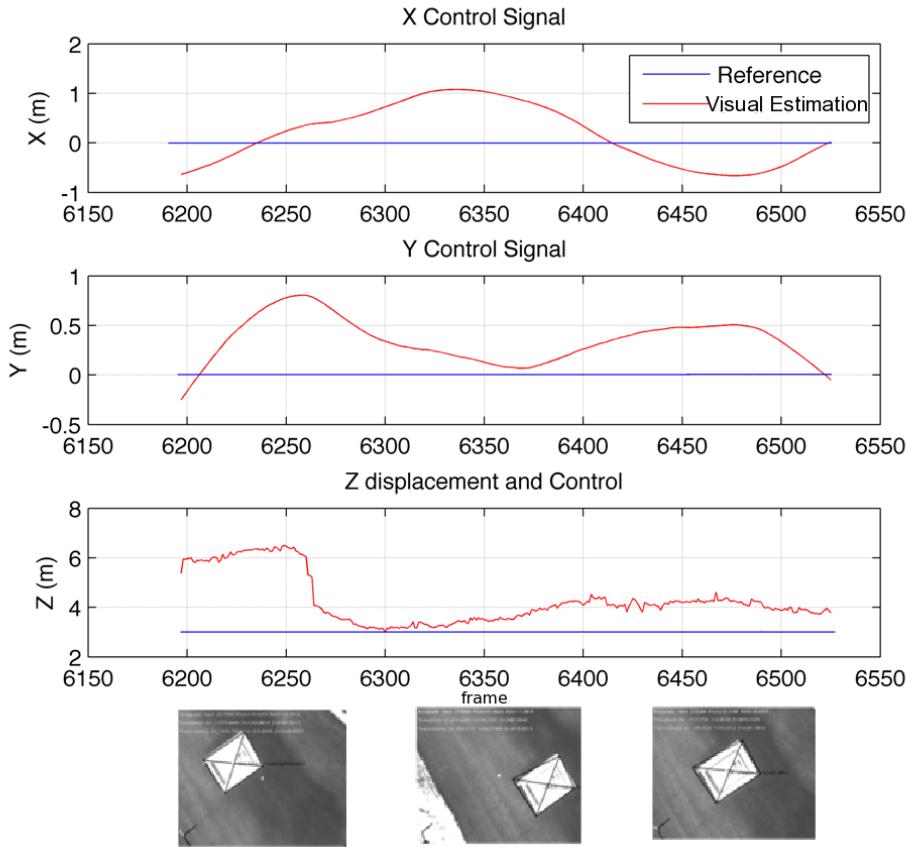


Figure 5.7: On-board UAV visual pose control. Position commands are sent to the flight controller to achieve the desired position [0 m, 0 m, 3 m]. Average frame rate is 12 Hz.

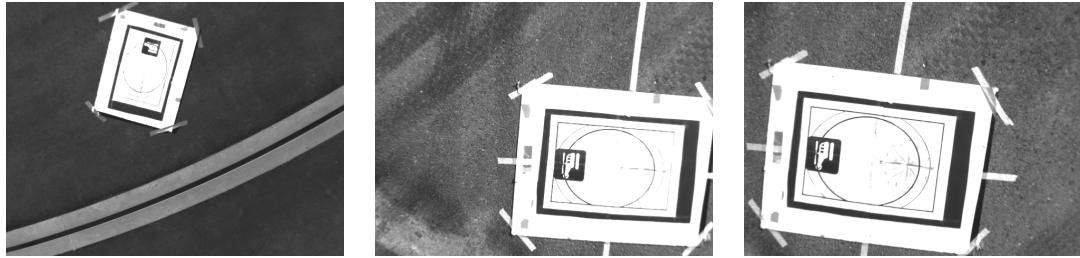


Figure 5.8: Helipad position on each of the three different landing tests. The helipad is oriented in such a way that X axis is North position, Y axis is East position and Z axis is Down Position (Earth center).

Figure 5.9 shows the reconstruction of the flight test 1, using IMU+GPS state estimated data. This Figure also shows the original reference image, the current frame, the optical flow between last and current frame, the helipad position in the camera coordinate system and the Tait-Bryan angles obtained from the rotation matrix. Landing test 1 begins when the UAV is hovering with

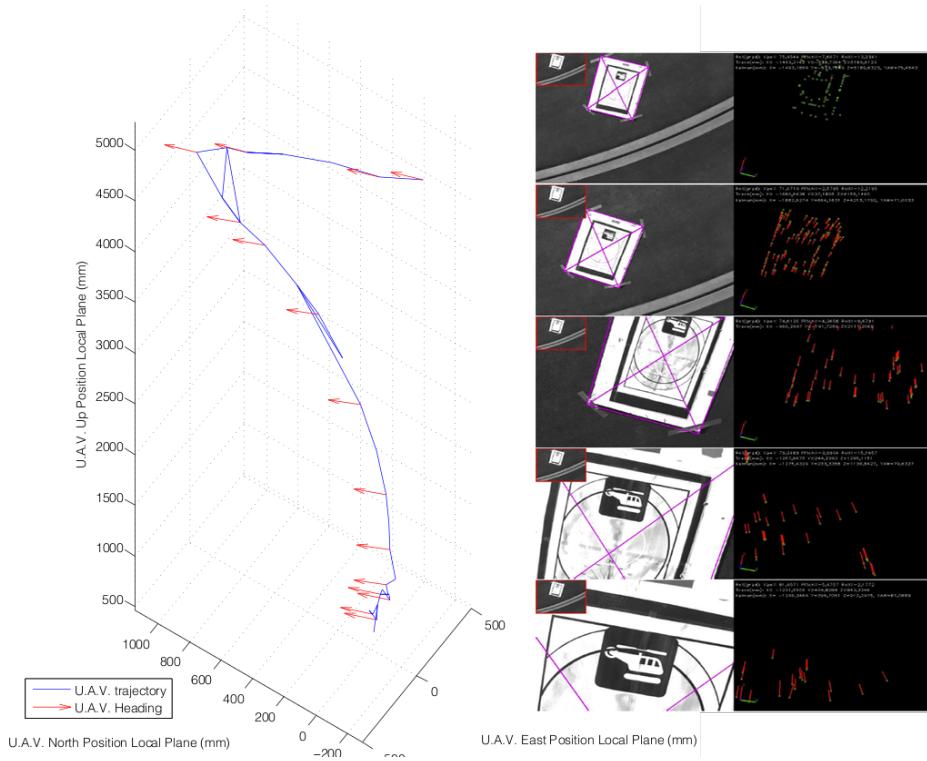


Figure 5.9: Landing process using a 3D pose estimation based on a helipad tracking. Landing test begins at an altitude of 5.2 m. Left: 3D flight and heading reconstruction for the landing test based on the IMU+GPS data. Right: On-board images showing the reference image I_0 , the optical flow between the actual and last frame. Superimposed is the projection of the original rectangle, the translation vector and the Tait-Bryan angles.

an altitude of 5.2 m. Then the controller uses the estimated 3D data to command the UAV. The on-board visual control systems runs with an average of 10 fps.

3D pose estimated using the visual system is compared with the helicopter IMU+GPS state estimated by the controller, with reference to the take-off point (Center of the Helipad). Because the local tangent plane to the helicopter is defined in such a way that the X axis is North position, the Y axis is East position and Z axis is Down Position (negative), the measured X and Y values must be rotated according to the helicopter heading or Yaw angle, in order to be comparable with the estimated values obtained from the homographies. Figures 5.10, 5.11 and 5.12 show the comparison between the IMU+GPS state estimated data and the estimated landmark position used for controlling the UAV. Figure 5.13, shows the estimated and measured inertial Yaw angle comparison.

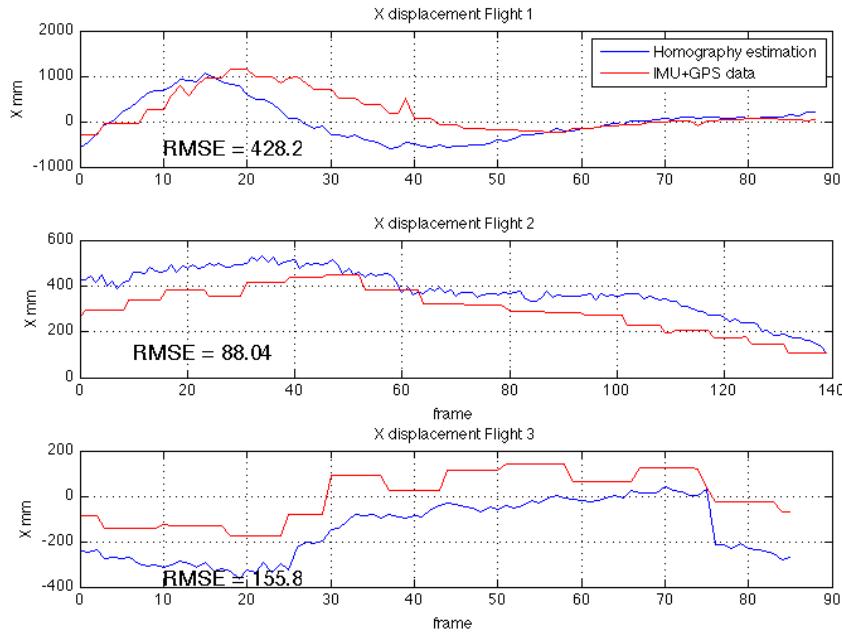


Figure 5.10: Comparison between the X axis displacement for homography estimation and IMU+GPS data. Average frame rate is 12 Hz

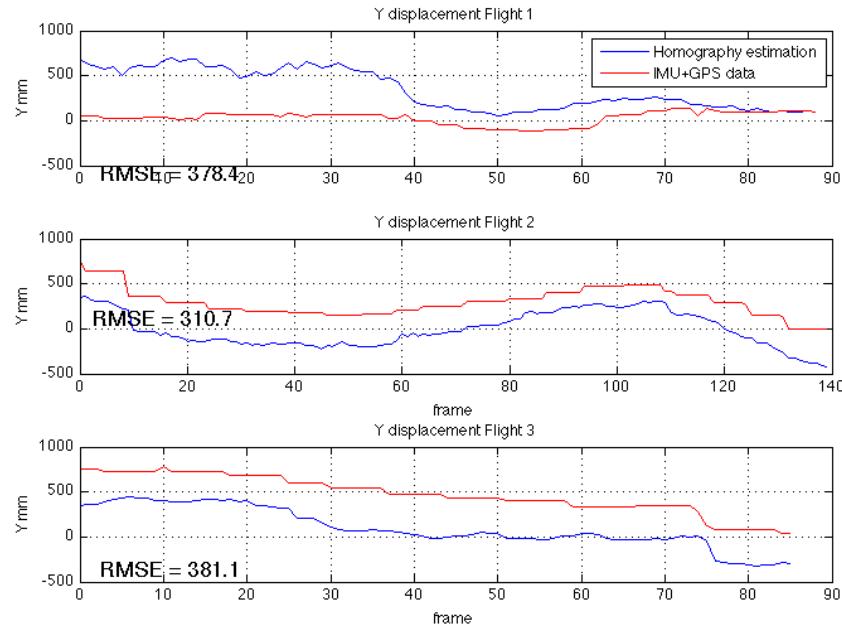


Figure 5.11: Comparison between the Y axis displacement for homography estimation and IMU+GPS data. Average frame rate is 12 Hz

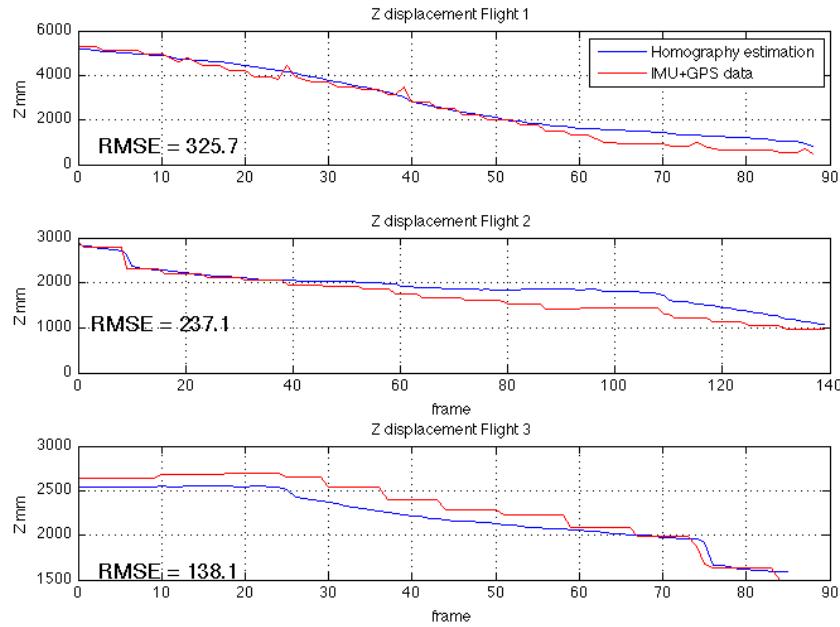


Figure 5.12: Comparison between the Z axis displacement for homography estimation and IMU+GPS data. Average frame rate is 12 Hz

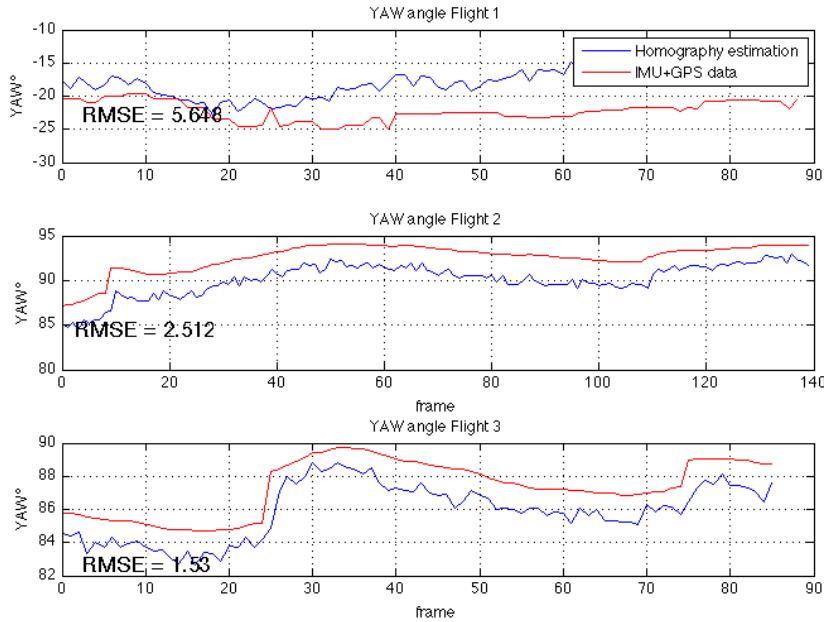


Figure 5.13: Comparison between the Yaw angle measured using homography estimation and IMU+GPS data. Average frame rate is 12 Hz

Results have shown a good performance of the implemented landing control. The estimated pose has been compared to the IMU+GPS state estimated data, showing a high degree of coherence. In all test, the UAV correctly descends and approaches the helipad. The visual control is not capable to accurately estimating the position of the helipad below the 0.7 m threshold, the constant descending command is thereby executed in order to carry out the final landing process and avoid ground effect. Video sequences of these landing tests are available on the Vision4UAV project Web Page (CVG-UPM, 2010)

It is difficult to compare the proposed hovering and landing method with other approaches present in the literature. It is caused by the use of different characteristics of the visual system, the different controllers and the different UAVs themselves. However, the method proposed by Merz (Merz et al., 2006) is closed to the one presented in this section. Both methods are based on the estimation of a helipad position based on homographies. The estimated pose is then used to control the UAV during a landing task.

Merz's method estimated the helipad position looking in every image for black circles forming an equilateral triangle with different resolution levels. These triangles are used for estimating the position of the helipad w.r.t camera. A visual servoing running in parallel is used for centering the landmark on the camera FOV by controlling a pan and tilt platform. Estimated pose is fused with the inertial measurements in order to have an accurate estimation. This information is then used to command the landing in a sequential control approach. The fusion of the visual information with the inertial data, provides of hight pose resolution. Results presented in (Merz et al., 2006), the RMSE has a stable behavior of few mm in simulated and real test.

The method presented on this section, only uses visual information without fuses it with other data. It causes that our pose estimation error is litter higher than other approaches based on sensor fusion. Nevertheless, the pose estimation has a better resolution that the one the aircraft autopilot is capable to control. Visual tracking based on features allows using this method with any kind of landmarks, while the camera remains fixed on-board. As experimental tests have demonstrated, the visual system is robust against occlusions up to 70 % of the helipad as well as changes in illuminations (Figure 4.18). This visual tracking and control systems only requires a rectangular helipad with known dimensions.

Proposed control system has an operating range similar such as other approaches. System correctly works above 0.5 m up to a height of 12 m. Below the 0.5 m threshold, visual landing methods are not able to identify the landmark. Moreover, the helipad tracking and the control of the aircraft are affected by the ground effect, when it is closed to the helipad. These issues are generally solved making a constant speed descending below the visual threshold.

5.4. Aerial object following based on visual information

In this section, a novel contribution for the task of aerial moving objects following is presented. The algorithm aims to follow a target T with an unknown trajectory. It has to be maintained in the camera FOV with a fixed separation distance. Considering Figure 5.14, the target object is a 3D spherical surface that is projected onto the image plane as a circular region that can be defined by its center of projection $\mathbf{x}_t = [x_t, y_t]^T$ and the circumference diameter ϕ_t . Because the target is an ideally spherical surface, the projection point (\mathbf{x}_t) can be considered as the image projection of target sphere centroid with coordinates on the camera frame defines as $\mathbf{X}_{Tc} = [X_{Tc}, Y_{Tc}, Z_{Tc}]^T$. The detected circumference (with ϕ_t pixels diameter) on the image plane corresponds to the projections of the sphere perimeter (with a fixed diameter \emptyset_T) that results of the intersection of the plane which is parallel to the vector defined by the camera optical centre and the target sphere centroid and divides the target sphere in two hemispheres. The projected diameter can also be used to measure the distance to the target, because it is inversely proportional to the distance from the camera to the object.

Following a similar approach to the one presented on section 5.2.3, it is possible to define the projected features $\mathbf{s} = [x_t, y_t, \phi_t]^T$ pixels as the images characteristics used to define the control law. Because the control goal is to maintain always the target object on the camera FOV with a fixed distance and considering that the projected diameter is inversely proportional to the target distance, the desired position on the image plane can be defined as $\mathbf{s}^* = [x^*, y^*, \phi^*]^T = [\frac{\text{ImgWidth}/2 - c_u}{f}, \frac{\text{ImgHeight}/2 - c_v}{f}, \frac{\phi^*}{f}]^T$, so the error function is defined as ($\mathbf{e} = \mathbf{s} - \mathbf{s}^*$).

Considering a general pinhole camera and in order to design a velocity controller, the features are projected onto the image plane as equation 5.16 shows:

$$\begin{aligned} x_t &= \frac{X_{Tc}}{Z_{Tc}} \\ y_t &= \frac{Y_{Tc}}{Z_{Tc}} \\ \phi_t &= \frac{\emptyset_T}{Z_{Tc}} \end{aligned} \quad (5.16)$$

The time derivatives of the projected target center and diameter (a fixed value) on the image plane are defined in Equation 5.17:

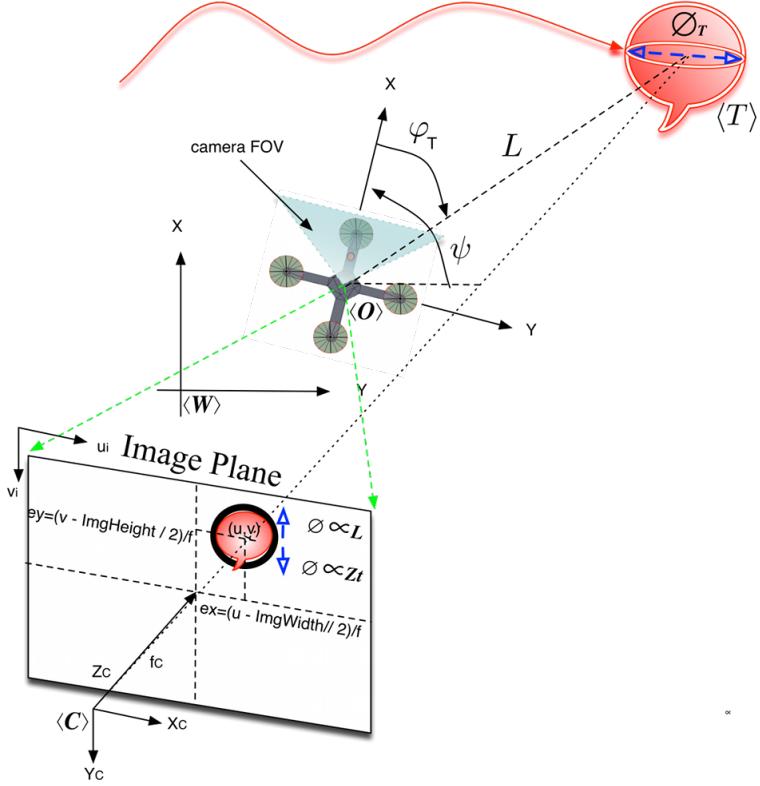


Figure 5.14: UAV object following setup in 2D. The control goal is to follow a target object (T) with an unknown trajectory using only the pose $((u, v)_t)$ and projection diameter (ϕ_t) on the image plane. The objective is to maintain the Target centered on the image plane with fixed space distance, i.e., a fixed proportional projection diameter on the image.

$$\begin{aligned}
 \dot{x}_t &= \frac{\dot{X}_{Tc}}{Z_{Tc}} - X_{Tc} \frac{\dot{Z}_{Tc}}{Z_{Tc}^2} = \frac{\dot{X}_{Tc}}{Z_{Tc}} - x_t \frac{\dot{Z}_{Tc}}{Z_{Tc}} \\
 \dot{y}_t &= \frac{\dot{Y}_{Tc}}{Z_{Tc}} - Y_{Tc} \frac{\dot{Z}_{Tc}}{Z_{Tc}^2} = \frac{\dot{Y}_{Tc}}{Z_{Tc}} - y_t \frac{\dot{Z}_{Tc}}{Z_{Tc}} \\
 \dot{\phi}_t &= -\phi_T \frac{\dot{Z}_{Tc}}{Z_{Tc}^2} = -\phi_t \frac{\dot{Z}_{Tc}}{Z_{Tc}}
 \end{aligned} \tag{5.17}$$

The time derivative of the point \mathbf{X}_{Tc} is defined in the same way as equation 5.3. Integrating it in Equation 5.17, and considering that features projection is defined as 5.16 shows, it is possible to obtain the integration matrix defined in Equation 5.18.

$$\mathbf{L}_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x_t}{Z_T} & x_t y_t & -(x_t^2 + 1) & y_t \\ 0 & -\frac{1}{Z} & \frac{y_t}{Z_T} & (y_t^2 + 1) & -x_t y_t & -x_t \\ 0 & 0 & \frac{\phi_t}{Z_T} & \phi_t y_t & -\phi_t x_t & 0 \end{bmatrix} \quad (5.18)$$

Equation 5.18 relates the 6 DOF camera movements with the target projected features on the image plane. Because there are only three features on the image plane, they constrain the number of camera DOF that can be controlled. In order to deal with these constraints the following assumptions are made:

- Vehicle *Roll* and *Pitch* are approximately constant, therefore camera angular velocities in *X* and *Z* are negligible. $\omega_{X_c} \approx \omega_{Z_c} \approx 0$
- Camera velocity on axis *X* is set to zero initially, i.e., $v_{X_c} = 0$, $t = 0$ however $v_{X_c} \neq 0$, $\forall t > 0$

using this assumptions, equation 5.18 can be written as:

$$\mathbf{L}_s = \begin{bmatrix} 0 & \frac{x_t}{Z_T} & -(x_t^2 + 1) \\ -\frac{1}{Z} & \frac{y_t}{Z_T} & -x_t y_t \\ 0 & \frac{\phi_t}{Z_T} & -\phi_t x_t \end{bmatrix} \quad (5.19)$$

The inverse of 5.19 is then as equation 5.20:

$$\widehat{\mathbf{L}}_e^+ = \begin{bmatrix} 0 & -Z & \frac{Z y_t}{\phi_t} \\ -Z x_t & 0 & \frac{Z(x_t^2 + 1)}{\phi_t} \\ -1 & 0 & \frac{x_t}{\phi_t} \end{bmatrix} \quad (5.20)$$

The velocity commands that will be sent to the low level controller are defined as in equation 5.21:

$$\begin{aligned} \mathbf{V}_c &= -\lambda \widehat{\mathbf{L}}_e^+ \mathbf{e} \\ \begin{bmatrix} v_y \\ v_z \\ \omega_y \end{bmatrix} &= -\lambda \begin{bmatrix} 0 & -Z & \frac{Z y_t}{\phi_t} \\ -Z x_t & 0 & \frac{Z(x_t^2 + 1)}{\phi_t} \\ -1 & 0 & \frac{x_t}{\phi_t} \end{bmatrix} \begin{bmatrix} x_t - x^* \\ y_t - y^* \\ \phi_t - \phi^* \end{bmatrix} \end{aligned} \quad (5.21)$$

Equations 5.19 and 5.20 show that there is a direct interaction between each one DOF of the camera and the features in the image. For control proposes, it is desirable to reduce the interaction between the different variables that allows obtaining a better tuning of the controller. In order to have a measure of the process interaction, the Relative Gain Array RGA defined as $\Lambda = L \otimes (L^{-1})^T$,

(where \otimes denotes element-by-element multiplication) is applied to the interaction matrix, given as a result the RGA presented in Equation 5.22.

$$\Lambda = \begin{bmatrix} 0 & -x_t^2 & x_t^2 + 1 \\ 1 & 0 & 0 \\ 0 & x_t^2 + 1 & -x_t^2 \end{bmatrix} \quad (5.22)$$

The RGA matrix 5.22 is evaluated for the steady state ($x_t = 0$) given as a result:

$$\Lambda = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (5.23)$$

The RGA matrix for the steady state shows that the system can be approximated by a series of "direct control" loops, in which the best input for the control of the Y_c axis velocity (v_y) is the y_t component of the projected point on the image plane. In the same way, the best input for the control of the Z_c axis velocity corresponds to the projected diameter on the image plane ϕ_t and finally ω_y is best controlled by the x_t component of the projected point on the image plane. In steady state, other pair interactions can be considered as negligible. Steady state interaction matrix 5.24 can be approximated as 5.24 shows.

$$\mathbf{L}_s = \begin{bmatrix} 0 & 0 & -(x_t^2 + 1) \\ -\frac{1}{Z} & 0 & 0 \\ 0 & \frac{\phi_t}{Z_T} & 0 \end{bmatrix} \quad (5.24)$$

The pseudo-inverse of the Estimated Jacobian $\widehat{\mathbf{L}}_e^+$ is presented in Equation 5.25.

$$\widehat{\mathbf{L}}_e^+ = \begin{bmatrix} 0 & -Z & 0 \\ 0 & 0 & \frac{Z}{\phi_t} \\ \frac{-1}{x_t^2 + 1} & 0 & 0 \end{bmatrix} \quad (5.25)$$

Equations 5.20 and 5.25 still needs to measure or estimate the depth of the target. However, the projected diameter on the image plane can be considered as being inversely proportional to the distance to the target and taking into account that the diameter range on the image varies from a minimum of one pixel to a maximum of $ImgWidth$ pixels, it is possible to make a linear approximation of the object distance given by $Z = \frac{k_Z}{\phi_t}$, where k_Z is a constant value that in the case of a pinhole camera can be defined as $k_Z = \mathcal{O}_T f$.

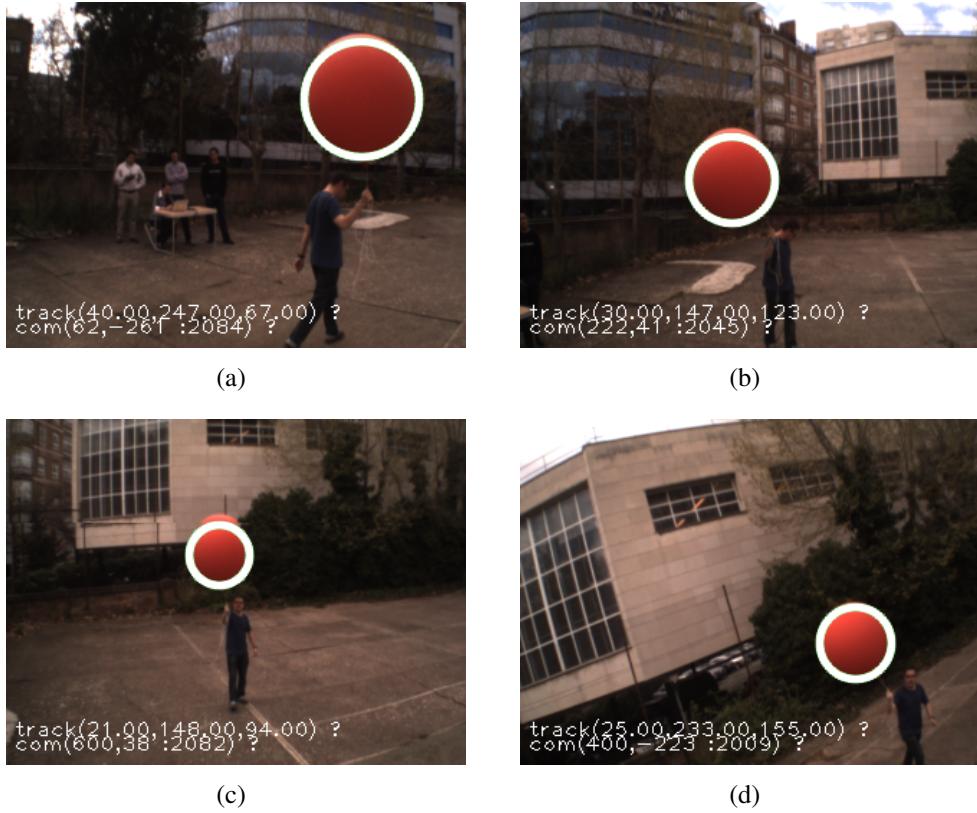


Figure 5.15: Camshift tracking of a colored red target on an image sequence. The white circle corresponds to the boundaries of the tracked colored area.

5.4.1. Color-based probabilistic tracking approach

We approach the problem of tracking by exploiting the color characteristic of the target. A basic color to the target is defined by assuming a simple colored mark and tracking this object. Therefore, a suitable and consistent color representation that allow color distributions derived from video image sequences that are approximately constant (in outdoor settings) is relied upon. However, this process is not always perfect, and changes still occur in color distributions over the time. An algorithm that has proven to deal with this issue by dynamically adapting to changes in probability distributions is the *Continuously Adaptive Mean Shift* Camshift (Bradski, 1998). This algorithm is based on the *mean shift* originally introduced by Fukunaga and Hostetler (Fukunaga and Hostetler, 1975).

The Camshift algorithm is used for tracking a defined color on image sequence, obtaining the center of the color region for each frame and the circumference that involves the tracked colored area. Figure 5.15 shows an example of a color tracking sequence using the Camshift algorithm on a red object.

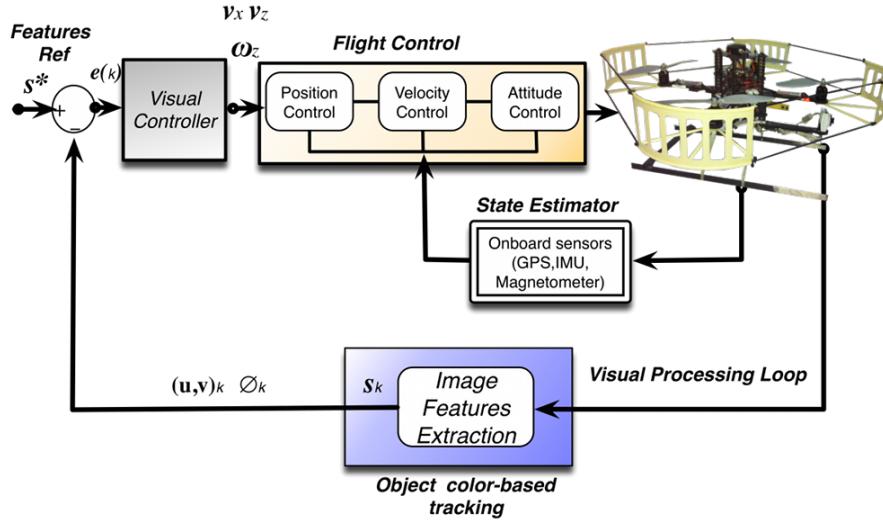


Figure 5.16: Object following *Dynamic image-based look-and-move* system architecture.

5.4.2. Tests and results

Each of the Image-based visual servoing matrixes defined in equations 5.20 and 5.25 are separately implemented on the Pelican UAV (appendix B), using a dynamic look-and-move architecture as presented in Figure 5.16. In this scheme, the velocity references generated by each of these controllers (running on-board aircraft) are used as input references for the Pelican low level controller. This low level controller allows receiving velocity commands inputs, as well as direct attitude and motor control actions. For this test, the velocity commands generated by the visual system are directly sent as input velocities for the Autopilot. This autopilot also allows controlling the X, Y, Z and Yaw independently. *Roll* and *Pitch* angles are no directly controlled by velocity command, but it is possible to control it by means of motor direct control. So, in this control architecture, and considering that the camera is looking forward, the quadrotor X and Z as well as the Yaw angle will be controlled by the generated references ($V_{Xq} = V_{Zc}$, $V_{Zq} = V_{Yc}$, $\omega_{Zq} = \omega_{Yc}$). The Y axis and *Pitch*, *Roll* angles are controlled by the low level autopilot.

Several tests have been conducted in which different balloons have been used as targets (in order to reduce the complexity of an aerial moving object). These balloons are moved with a random trajectory on the 3D space. The visual system composed of the Camshift color tracker and the IBVS proposed, runs on-board the aircraft in an Intel Atom Board PC 1.6 GHz dual core with an average of 20 fps. The velocity commands are sent to the low level autopilot through a serial interface. Figure 5.17 shows an external view of two different

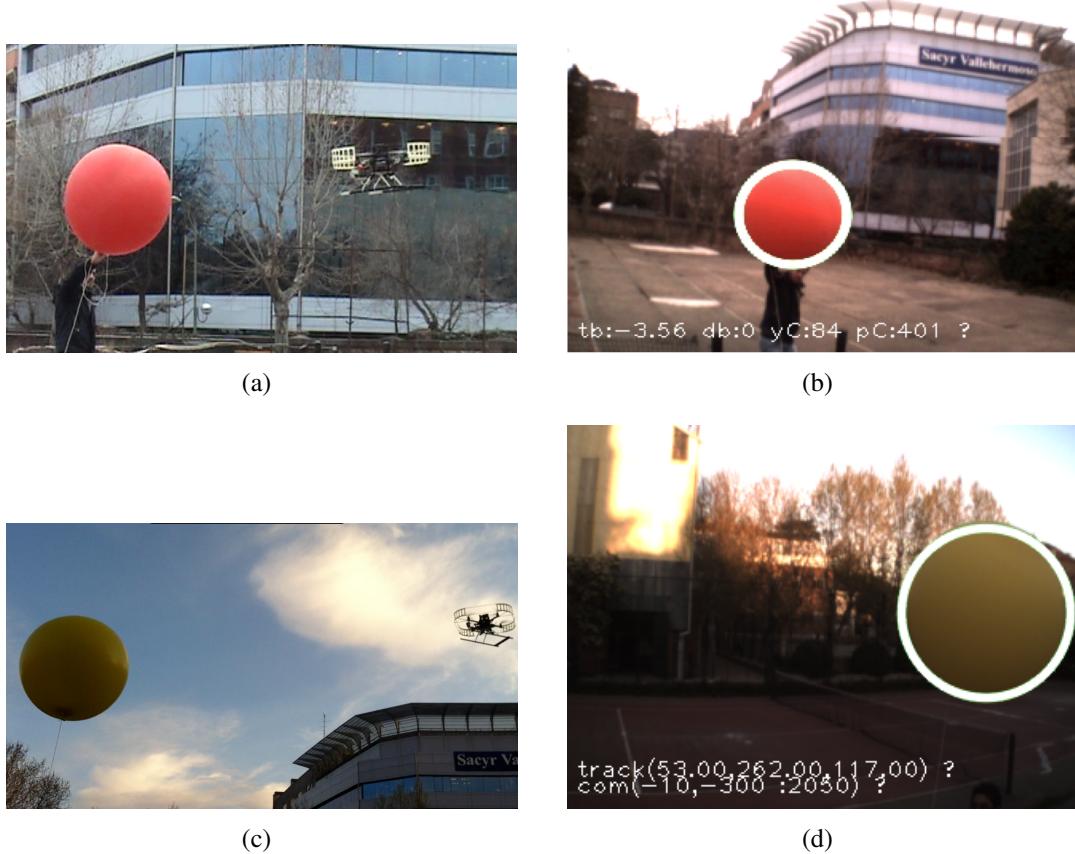


Figure 5.17: Two different object following tests using an Image-based visual servoing IBVS on a Pelican quadrotor. Average frame rate is 20 Hz. (a),(c): External view of the tests using two balloons with a different color. The balloons are manually move with a random trajectory. (b),(d): The on-board image captured, and the projected diameter on the image plane identified using the Camshift color tracker.

object following tests and the on-board images used for the IBVS.

Both controllers (full matrix and reduced version) gains are tuned using step references with the balloons on static position. In this tests, the low level controller is commanded by the references generated by the IBVS controller. Figure 5.18 shows the response of the controllers presented in equations 5.20 and 5.25 to step changes on u^* with the balloon being on a static position (some noise is caused by low wind, making the balloon oscillate around the anchor point).

In the same way, Figure 5.19 shows the response of the controllers presented in equations 5.20 and 5.25 to step changes on ϕ^* with the balloon on a static

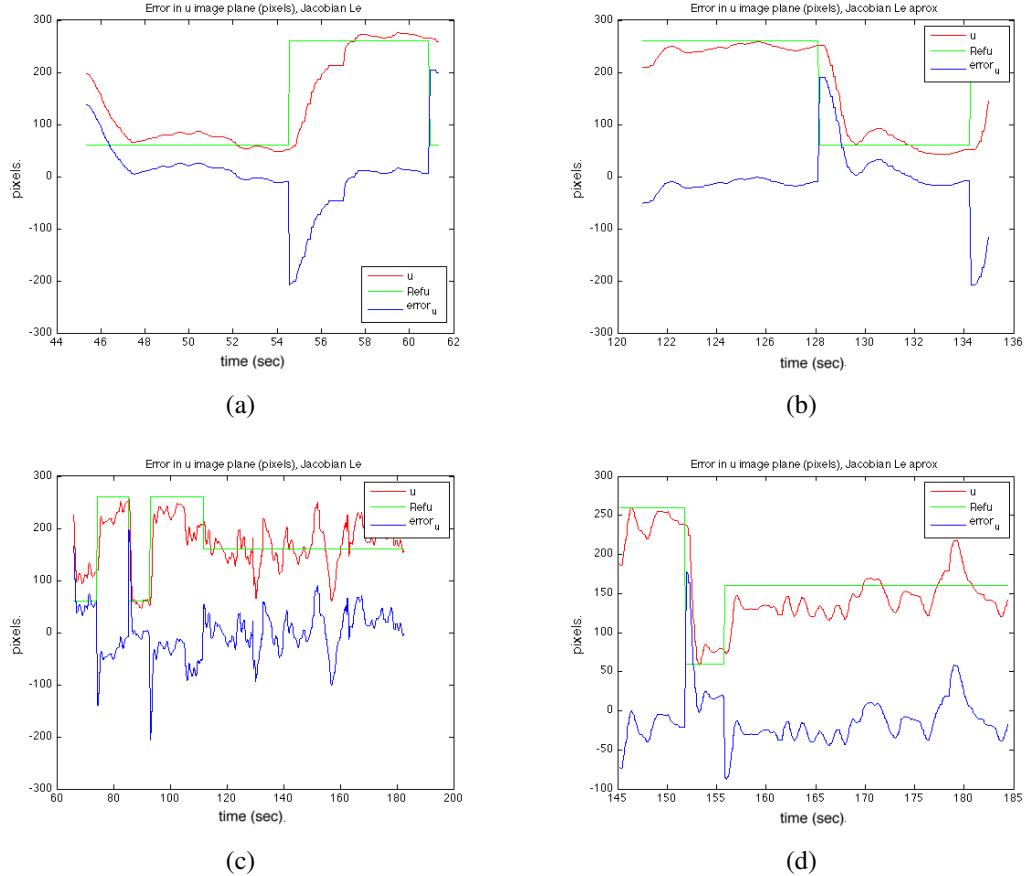


Figure 5.18: $\widehat{\mathbf{L}}_e^+$ IBVS step response on u^* change. Average frame rate is 20 Hz. (a),(c): Two test using the full version of the IBVS presented in Equation 5.20. (b),(d): Two test using the approximated version of the IBVS presented in equation 5.25.

position.

Figure 5.20 shows the 2D reconstruction of the flight test presented in Figures 5.18(d) and 5.19(b). Superimposed images correspond to captured images from the on-board system. The Camshift color tracking is used in order to track the balloon projection on the image plane and then used to estimate its position and diameter projection on the image plane, from which the IBVS commands are derived.

The controllers have also been tested with a moving target. The test begins when the visual systems correctly identify and track the balloon on the image plane, then the balloon is moved with a random trajectory on the 3D space and consequently, the command generated by the IBVS controller is executed by

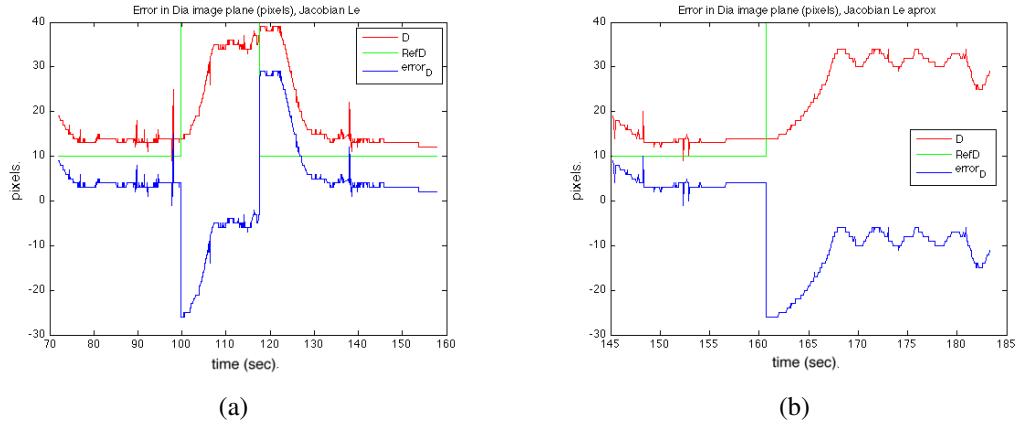


Figure 5.19: $\widehat{\mathbf{L}_e}^+$ IBVS step response on ϕ^* change. Average frame rate is 20 Hz.
 (a): test using the full version of the IBVS presented in Equation 5.20. (b): test using the approximated version of the IBVS presented in Equation 5.25. (this test corresponds with the test presented in Figure 5.18(d)).

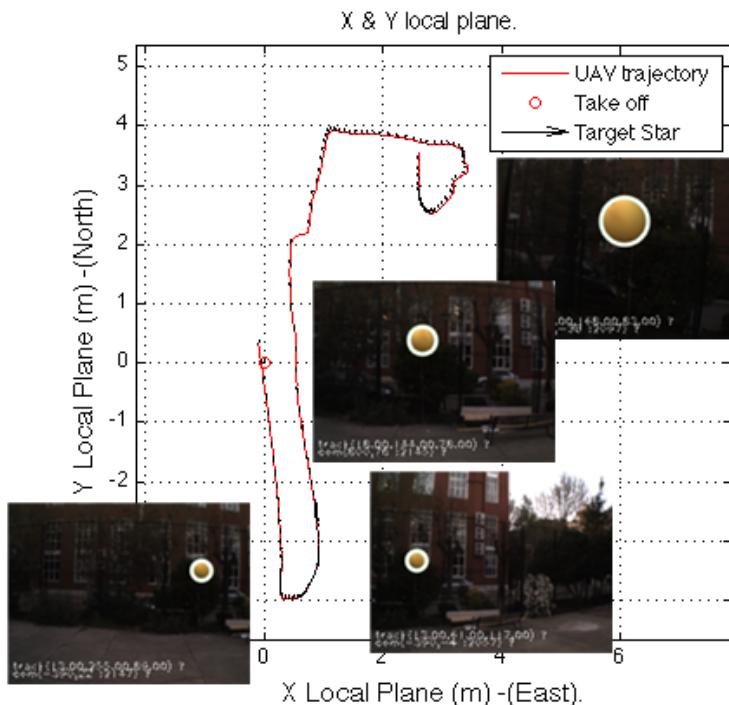


Figure 5.20: 2D reconstruction for test with steps changes on u^* and ϕ^* presented in Figures 5.18(d) and 5.19(b). This step test was done with an anchor balloon. First a reference step is applied on u^* , then the step change is applied on ϕ^* . Superimposed images show the captured images at different moments of this test.

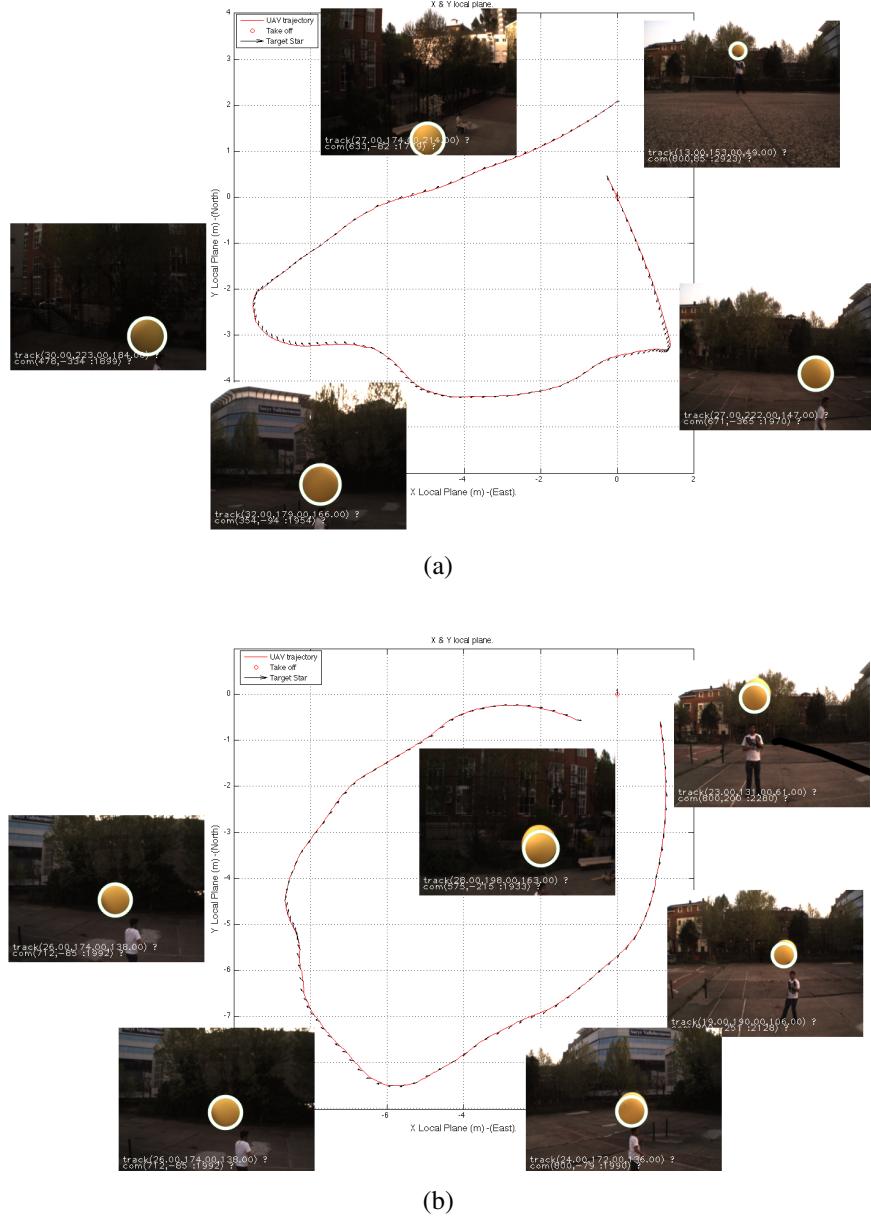


Figure 5.21: 2D reconstruction for Object following test. (a): Complete interaction matrix $\widehat{\mathbf{L}}_e^+$. (b): approximated interaction matrix $\widehat{\mathbf{L}}_e^+$.

the low level controller. Figure 5.21 presents two different tests employing the complete and approximated interaction matrix.

Figures 5.22, 5.23 and 5.24, shows the IBVS response given by both controllers for the test presented in Figure 5.21.

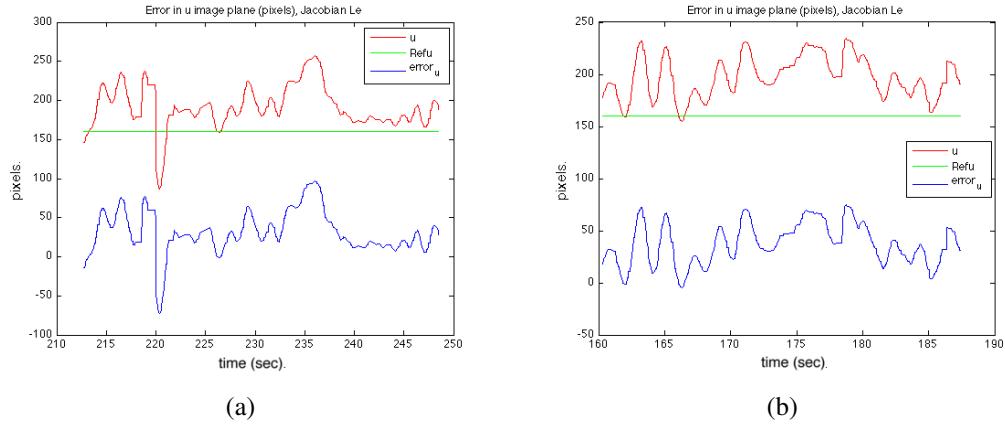


Figure 5.22: $2\widehat{\mathbf{L}}_e^+$ IBVS response on u^* for a moving target following test. Average frame rate is 20 Hz. (a): Complete interaction matrix $\widehat{\mathbf{L}}_e^+$. (b): approximated interaction matrix $\widehat{\mathbf{L}}_e^+$.

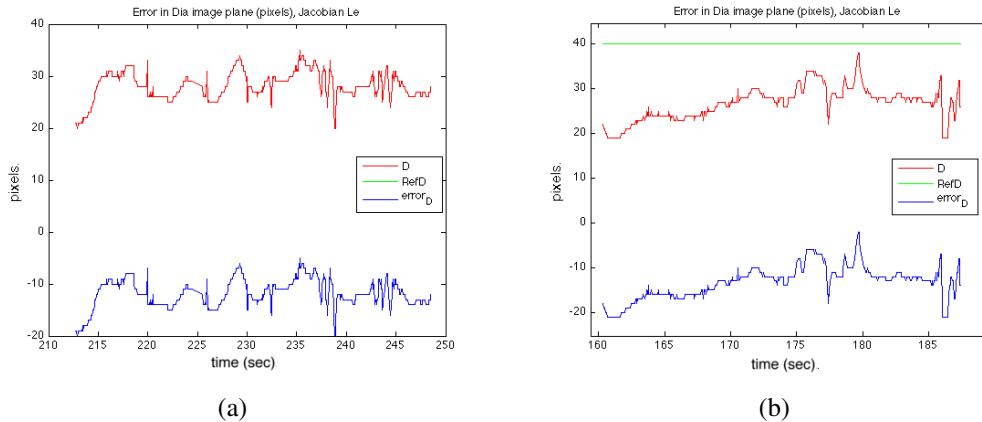


Figure 5.23: $2\widehat{\mathbf{L}}_e^+$ IBVS response on u^* for a moving target following test. Average frame rate is 20 Hz. (a): Complete interaction matrix $\widehat{\mathbf{L}}_e^+$. (b): approximated interaction matrix $\widehat{\mathbf{L}}_e^+$.

Tests have shown that there is not an important difference between the full IBVS interaction matrix presented in equation 5.20 and the approximated Jacobian presented in equation 5.25. According to this results, it is also possible to assume that the different degrees of control can be decoupled, allowing to design "direct control" schemes separately for each of the involved DOF (exception of *Roll* and *Pitch* that are controlled directly by the aircraft Autopilot).

A simple method used for estimating the object depth, based on the projected

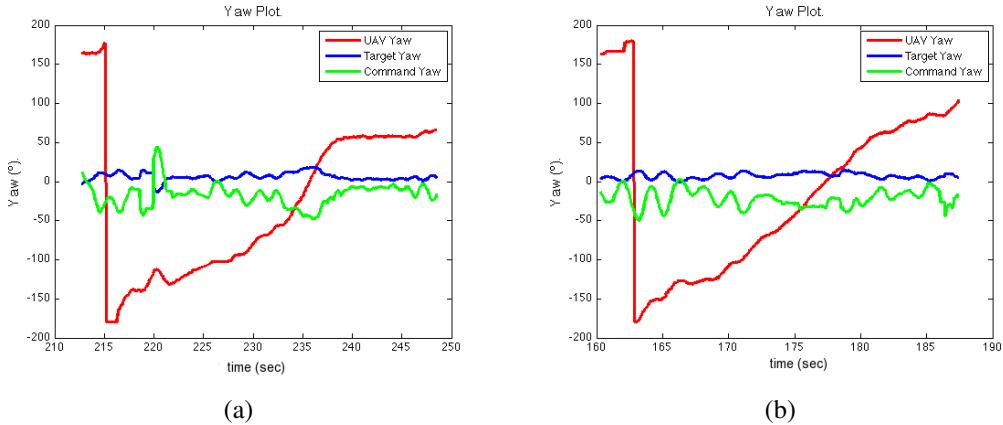


Figure 5.24: $2\widehat{\mathbf{L}}_e^+$ IBVS Yaw for a moving target following test. Average frame rate is 20 Hz. (a): Complete interaction matrix $\widehat{\mathbf{L}}_e^+$. (b): approximated interaction matrix $\widehat{\mathbf{L}}_e^+$.

diameter on the image plane have been proposed. It has demonstrated being a good approximation that allows generating an operative IBVS. Both controllers were tested again step changes in set points references, as well as follow an object with an undefined trajectory (including perturbations caused by wind), showing the robustness of the presented control system.

5.5. Conclusions

This chapter has presented how the classical visual servoing architectures (Image-based IBVS and Position based visual servoing PBVS) are exploited in order to develop novel control applications for UAV. A review of classical control architectures was firstly introduced and then used in order to define a dynamic look-and-move architecture that allows generating a hierarchical control structure where high level controller is composed by the visual processing systems that generates position or velocity references for a low level control unit.

Two visual control systems were designed using the proposed high level decoupled visual servoing architecture. These strategies allow relating controlled variables directly with a unique input feature, generating direct and simple linear controls. The proposed methods have shown an excellent robustness to drive the non-holonomic behavior of rotary wing vehicles, as well as the external perturbations commonly presented in outdoors flights

A position based visual servoing was successfully implemented in order to generate a hovering positioning and landing controller based on 3D position estimated with and homography based method. The systems have been

implemented and tested on rotary wing helicopter in order to make accurate positioning and landing based on the tracking of helipad. The 3D positions estimated by the visual systems directly allows generating a PBVS control loop with a decoupled behavior between the translational and rotational motions.

Finally, an image based visual servoing was derived based on the projected characteristics of a moving target on the image plane and its tracking based on a probabilistic color analysis algorithm. The detected target features on the image plane are used to develop a purely IBVS scheme. The steady state response of this controller was contemplated in order to generate an approximated decoupled version on the three controlled DOF. Both controllers were tested under the same conditions, showing that the approximated decoupled controller has a similar behavior to the one given by the full controller derived by using the classical IBVS approach.

Chapter 6

Omnidirectional Vision for UAVs

6.1. Introduction

In this chapter, omnidirectional vision for on-board UAVs is described. Two novel contributions for Omnidirectional systems in aerial robotics are presented. The first contribution is an attitude sensor for simultaneous *Roll*, *Pitch* and relative heading or *Yaw* angle measurement. The second is the development of a See&Avoid application, based on the properties given by omnidirectional systems and IBVS techniques.

Section 6.2 introduces the unified projection model used for characterizing most of omnidirectional systems, including catadioptric and dioptric systems. Section 6.3 shows an image processing algorithm to isolate the skyline from the catadioptric image implemented with the aim to estimate the attitude of the bodyframe. Relative *Yaw* or heading is also obtained using the so called visual compass on appearance images. The feasibility and reliability of this approach is presented when the obtained results are compared with the Inertial Measurement Unit IMU of the UAV testbed employed.

Finally, in section 6.4 the unified model is used in a dioptric system for a contribution in the field of See&Avoid for light UAV. Real flight test result are presented.

6.2. Central Catadioptric Cameras and Omnidirectional Images

This section is a short review and introduction for the design, construction and properties of Central Catadioptric Cameras and its implications for image processing. Catadioptric Cameras are devices that combined reflective elements (catoptric) and refractive systems (dioptric) to form a projection onto the image plane of the camera. They can be classified as Central and NonCentral Catadioptric cameras according to the single effective viewpoint criteria. Baker and Nayar (Nayar and Baker, 1997), (Baker and Nayar, 1999), define the configurations that satisfy the constraints of a single viewpoint, finding that a central catadioptric system can be built combining a perspective camera with a hyperbolic, elliptical or planar mirror, or using an orthographic camera with a parabolic mirror. The single view point constraint is achieved always by a planar mirror and a perspective camera, while for the hyperbolic and elliptical systems the uniqueness of the effective view point is satisfied if the center of the perspective camera is coincident with one of mirror focus. In the case of a parabolic mirror, the constraint is satisfied whenever the camera is orthographic and its optical axis is aligned with the paraboloid axis without limitations in the camera distance to the mirror. Figure 6.1 shows the four catadioptric systems that have satisfied the single view constrain.

Baker a Nayar also defined two degenerate configurations that combine a perspective camera with a spherical or conical mirror as shown in Figure 6.2. This degenerate configurations cannot be used for constructing useful cameras with a single effective view point. In these cases, a sphere is the limit of an ellipse when the two focal points coincide causing to place the camera in the center of the sphere (inside) to verify the single viewpoint property, meaning that the camera would see only itself. In the same way, a conical mirror is the limit of a pinhole camera, causing it to place the camera at the vertex of the cone, meaning that the camera would see nothing.

Geyer and Daniilidis (Geyer and Daniilidis, 2000), (Geyer and Daniilidis, 2001) proposed a unified model for the projective geometry induced by central catadioptric systems, showing that these projections are isomorphic to a projective mapping from a sphere (centered on the effective viewpoint) to a plane with the projection center in the perpendicular axis to the plane. In this model, a unit sphere is centered at the origin C . This model allows a reduced representation with two parameters, l and m , representing the distances from the sphere center to the projection center, O and to the plane as shown in Figure 6.3.

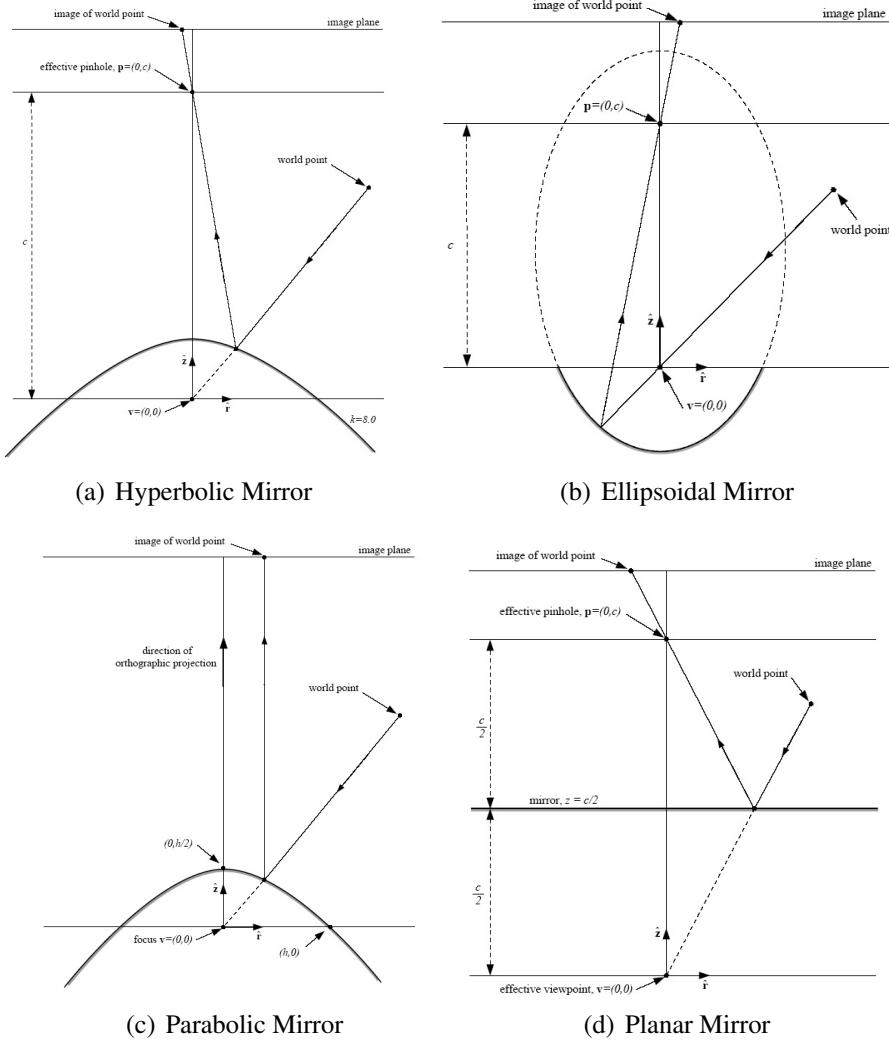


Figure 6.1: Four catadioptric systems that satisfied the single effective viewpoint constrain. Images originally published in (Baker and Nayar, 1999).

The projection of a point in space $[x, y, z]^t$ to an image point $[u, v]^t$ can be written as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{l+m}{l \cdot r - z_w} \begin{bmatrix} x_w \\ y_w \end{bmatrix} = \mathcal{P}(x_w, y_w, z_w; k, m) \quad (6.1)$$

$$r = \sqrt{x_w^2 + y_w^2 + z_w^2}$$

A modified version of this unified model is presented by Barreto and Araujo in (Barreto and Araujo, 2001), (Barreto and Araujo, 2002), where the mapping between points in the 3D world and points in the catadioptric image plane is split into three steps. First, a linear function maps the world into an oriented projective plane. Then a non-linear function transforms points between two oriented

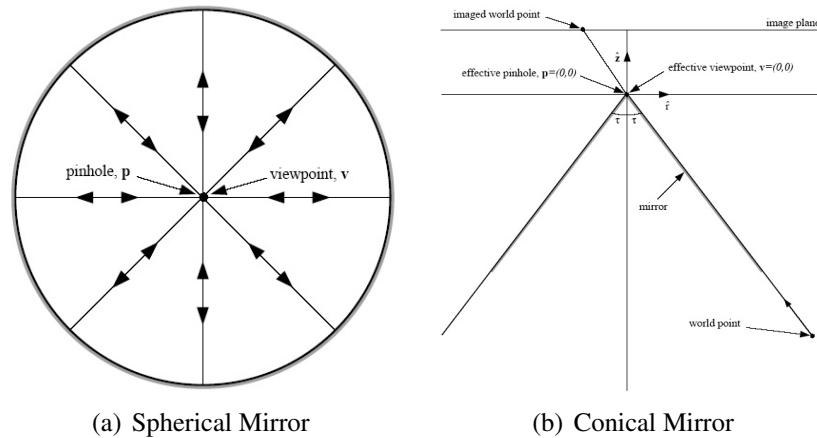


Figure 6.2: Two degenerated configurations combining either, a spherical or conical mirror and a perspective camera.. Images originally published in (Baker and Nayar, 1999).

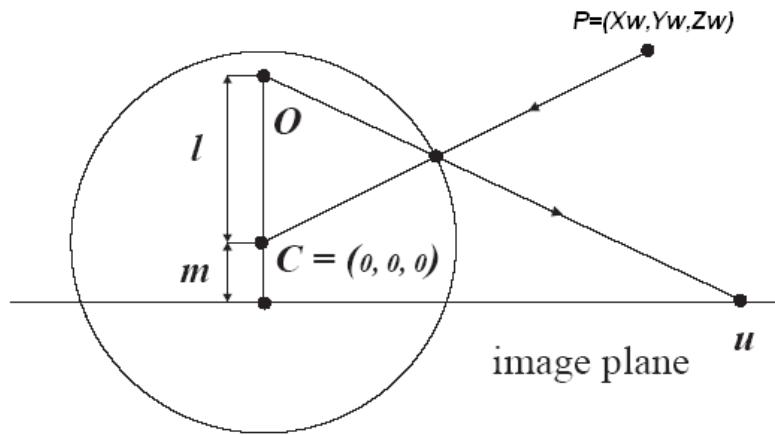


Figure 6.3: Gayer Unitary Sphere Projection Model to the plane.

projective planes. Finally there is a collineation function depending on the mirror parameters and the camera calibration matrix (intrinsic parameters). Figure 6.4 shows the general unit sphere projection for modeling catadioptric systems.

Consider a point in space (visible to the catadioptric system), with Cartesian coordinates $\mathbf{X}_w = [x_w, y_w, z_w]^T$ in the catadioptric reference (focus). This point is mapped onto point $\mathbf{X}_s = [x_s, y_s, z_s]^T$ on the unitary sphere centered on the effective view point by equation 6.2.

$$\mathbf{X}_s = \frac{\mathbf{X}_w}{\sqrt{x_w^2 + y_w^2 + z_w^2}} = \frac{\mathbf{X}_w}{\|\mathbf{X}_w\|} \quad (6.2)$$

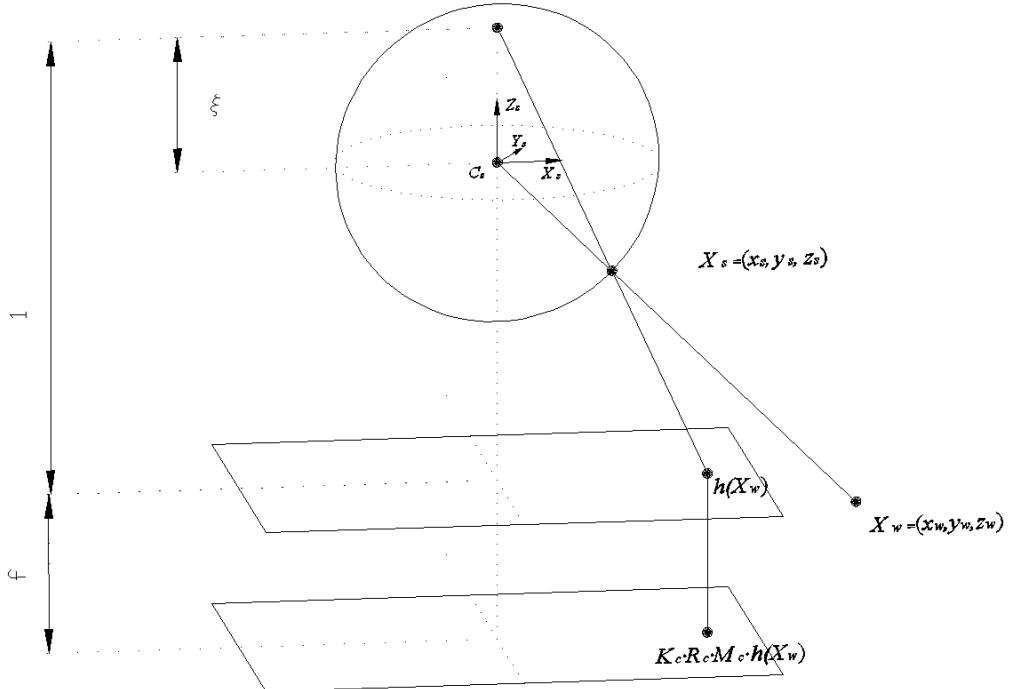


Figure 6.4: Catadioptric projection modeled by the unit sphere.

To each projective point \mathbf{X}_s , corresponds a projective point $\mathbf{X}_c = [x_c, y_c, z_c]^T$ in a coordinate system with origin at the camera projection center. This projection is a non-linear mapping between two projective planes and is defined by equation 6.3.

$$\mathbf{X}_c = [x_c, y_c, z_c]^T = \mathbf{M}_c \cdot \mathbf{h}(\mathbf{X}_w)$$

where

$$\mathbf{M}_c = \begin{bmatrix} \psi - \xi & 0 & 0 \\ 0 & \xi - \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.3)$$

$$\mathbf{h}(\mathbf{X}_w) = \left[x_w, y_w, z_w + \xi \sqrt{x_w^2 + y_w^2 + z_w^2} \right]^T$$

where the matrix \mathbf{M}_c depends on the mirror parameters ξ and ψ , defined for each one of the central catadioptric projections, as shown in Table 6.1.

Finally, the image in the catadioptric plane is obtained after a collineation between the image and the projective plane depending of the camera's intrinsic parameters \mathbf{K}_c (where m_x and m_y are the pixels per unit distance in image coordinates, f is the focal distance and (x_0, y_0) are the coordinates of the

Table 6.1: Parameters ξ and ψ for central catadioptric systems (d is the distance between focus and $4p$ is the Lactus Rectum.)

	Parabolic	Hyperbolic	Elliptical	Planar
ξ	1	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{d}{\sqrt{d^2+4p^2}}$	0
ψ	$1+2p$	$\frac{d+2p}{\sqrt{d^2+4p^2}}$	$\frac{d-2p}{\sqrt{d^2+4p^2}}$	1

principal point), and the rotation of the camera \mathbf{R}_c . The projection of a world point on the catadioptric image is defined by equation 6.4.

$$\begin{aligned}\mathbf{H}_c &= \mathbf{K}_c \mathbf{R}_c \mathbf{M}_c \\ \mathbf{X}_i &= \mathbf{H}_c \mathbf{h}(\mathbf{X}_w) \\ \mathbf{K}_c &= \begin{bmatrix} fm_x & s & x_0 \\ 0 & fm_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned}\tag{6.4}$$

$\mathbf{h}(\mathbf{X}_w)$ is a homogenous positive injective function, with an inverse defined by $\mathbf{h}^{-1}(\mathbf{X}_w)$. This function maps points in a projective plane onto the unitary sphere. The non-linear inverse function is defined as 6.5 shows.

$$\begin{aligned}[x_s, y_s, z_s]^t &= \mathbf{h}^{-1}(\mathbf{H}_c^{-1} \mathbf{X}_i) = [\lambda_c x_c, \lambda_c y_c, \lambda_c z_c - \xi]^t \\ \text{where} \\ \lambda_c &= \frac{z_c \xi + \sqrt{z_c^2 + (1 - \xi^2)(x_c^2 + y_c^2)}}{x_c^2 + y_c^2 + z_c^2}\end{aligned}\tag{6.5}$$

Once a catadioptric image was back-projected onto the unitary sphere, it is possible to obtain perspective images as if acquired by perspective camera by re-projecting to a new desired image plane from the sphere center. Assuming that the new virtual perspective camera is located in the center of the sphere, a rotation matrix \mathbf{R}_v , a desired intrinsic parameters of the virtual camera \mathbf{K}_v and a scaling factor λ_v are defined. The new virtual perspective image is obtained using equation 6.6.

$$\begin{bmatrix} u_v \\ v_v \\ 1 \end{bmatrix} = \lambda_v \mathbf{K}_v \mathbf{R}_v \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}\tag{6.6}$$

6.2.1. Mirror Design

Svoboda, Pajdla and Hlaváč (Svoboda et al., 1997), (Svoboda and Pajdla, 2002), developed a projection model for a perspective camera with a hyperbolic

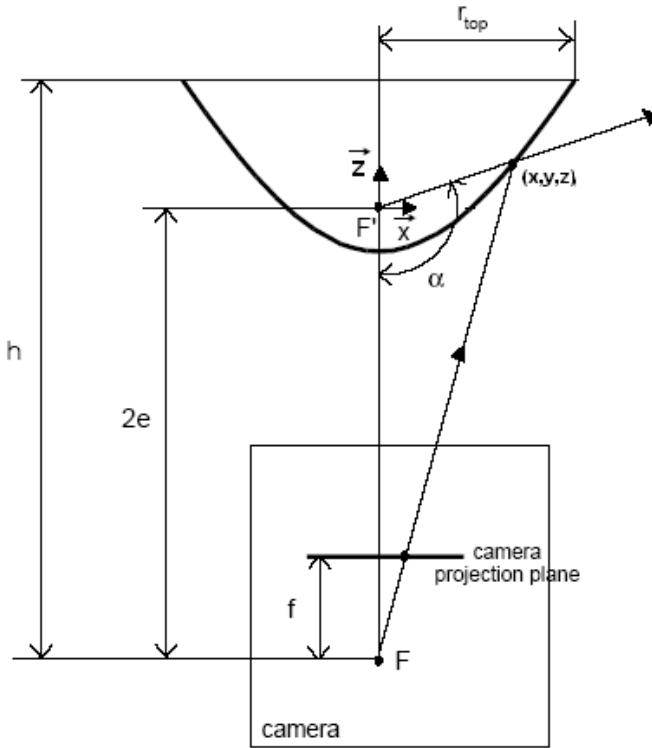


Figure 6.5: Svoboda *et al.* model for a perspective camera with a hyperbolic Mirror.

mirror and gave a general approach to the design and construction of a useful catadioptric system for mobile robots. They also explain the appropriate assembly and alignment of the camera and mirror in order to obtain the adequate images with central projection center. This approach assumes a hyperbolic shaped mirror centered in one of its focal points F' whose general function is defined by equation 6.7:

$$\left(\frac{z + \sqrt{a^2 + b^2}}{a}\right)^2 - \left(\frac{x}{b}\right)^2 - \left(\frac{y}{b}\right)^2 = 1 \quad (6.7)$$

Where a, b are the mirror parameters. The mirror eccentricity is $e = \sqrt{a^2 + b^2}$ and the distance between focus is defined as $d = 2e$.

Figure 6.5 shows the projection model and mirror design parameters used by Svoboda.

The two main requirements have to be considered during the design of the mirror's shape: The camera mirror has to be compact because it is intended for a mobile robot, and the system would have a large view angle. The maximum

view angle takes into account the dimension of the mirror to a distance where the camera is:

$$\alpha = \frac{\pi}{2} + \arctan\left(\frac{h-2e}{r_{top}}\right) \quad (6.8)$$

where r_{top} is the radius of any point on the mirror top rim ($r_{top}^2 = x_{top}^2 + y_{top}^2$), and h is the distance between the top of the mirror and the camera center, that is obtained using the equation 6.7 and translating it to the focal Point F by the eccentricity e as shown in Equation 6.9:

$$\begin{aligned} z_{top} &= \frac{a}{b} \sqrt{b^2 + r_{top}^2} - \sqrt{a^2 + b^2} \\ h &= z_{top} + 2e \end{aligned} \quad (6.9)$$

$$h = \sqrt{a^2 + b^2} + \frac{a}{b} \sqrt{b^2 + r_{top}^2}$$

Any point on the mirror rim $\mathbf{X}_{rim} = [x_{rim}, y_{rim}, h]^t$ referenced to the camera center is projected onto the image plane in a point $[x_{pix_{rim}}, y_{pix_{rim}}]$, by a perspective transformation $\mathbf{X}_{img} = \mathbf{K}_{cam}\mathbf{X}_{rim}$ where \mathbf{K}_{cam} is the camera calibration matrix. In this way, the circular mirror rim with radius r_{rim} is projected onto the image plane to a circle with radio $r_{pix_{rim}} = \sqrt{x_{pix_{rim}}^2 + y_{pix_{rim}}^2}$ pixels, depending on the camera calibration parameters:

$$\begin{aligned} r_{pix_{rim}}^2 &= x_{pix_{rim}}^2 + y_{pix_{rim}}^2 \\ r_{pix_{rim}}^2 &= \frac{(\mathbf{K}_{(1,1)}^2 x_{rim})^2}{h^2} + \frac{(\mathbf{K}_{(2,2)}^2 y_{rim})^2}{h^2} \end{aligned} \quad (6.10)$$

where $\mathbf{K}_{(1,1)}, \mathbf{K}_{(2,2)}$ are the scaled focal-x, focal-y parameters respectively of the camera. Assuming that the camera has a squared pixel construction ($\mathbf{K}_{(1,1)} = \mathbf{K}_{(2,2)}$) the distance from the camera center to the top of the mirror is a function of the ratio between the mirror radius and the desired radius of the rim projection on the image plane as indicate equation 6.11.

$$h = \mathbf{K}_{(1,1)} \cdot \frac{r_{rim}}{r_{pix_{rim}}} \quad (6.11)$$

The design goal is to obtain a projection in which the image of the mirror rim occupies the major part of the whole image to attain a best resolution as possible. To accomplish this objective two steps have to be made:

1. Define a feasible distance between camera and mirror top h , the rim radius r_{rim} and its desired projection radius on the image plane $r_{pix_{rim}}$.
2. Select a camera that satisfies equation 6.11

Once the distance from the camera to the top of the mirror is defined, the mirror shape is calculated. Assume that any point on the mirror rim, referenced to mirror focal point F' ($z_{rim} = h - 2e$) has to satisfy the equation of the mirror shape. Substituting the point on the equation 6.5 we get:

$$\frac{(h - \sqrt{a^2 + b^2})^2}{a^2} - \frac{r_{rim}^2}{b^2} = 1 \quad (6.12)$$

Equation 6.12 establishes a relation between the height of the system and the mirror radius (already defined) and the mirror shape parameters a, b . To obtain this parameters, this equation can be solved to the mirror parameter b obtaining that the only realizable ($z_{rim} > 0$) solutions is:

$$b = h\sqrt{\frac{a^2}{b^2} + 1} - \frac{a}{b}\sqrt{h^2 + r^2} \quad (6.13)$$

Equation 6.13 establishes a relation between the parameters a, b that defines the mirror shape. Svoboda *et al.* (Svoboda et al., 1997) shows that a suitable range for this ratio is $2 < a/b < 3$ where a low ratio gives a good resolution in the area close to the horizontal view and a large ratio increases the field of view but reduce the resolution.

Final Designed Mirror

The employed catadioptric system is based on the hyperbolic shape proposed by Okamoto *et. al* (Okamoto Jr and Grassi Jr., 2002), designed using the Svoboda method (Svoboda et al., 1997). It has a height from the top mirror to camera center of $h = 100$ mm and the radius of the mirror is set to $r_{rim} = 20$ mm. A standard firewire 30 fps, CCD camera with resolution of 640x480 pixels is selected and looking for the maximum occupancy on the image, the projected radius is defined to $r_{pix_{rim}} = 240$ pixels. Assuming a pixel size of 0.01 mm, a lens with focal distance of $f = 12$ mm has to be used.

Finally, defining a ratio $a/b = 2$ and replacing in Equation 6.13 we find that the mirror shape parameters are:

$$\begin{aligned} b &= 100\sqrt{4+1} - 2\sqrt{100^2 + 20^2} \\ b &= 19.6461 \\ a &= 2b = 39.2922 \end{aligned} \quad (6.14)$$

Figure 6.6 shows the final design and camera mirror distribution. A transparent extruded thermoplastic acrylic tube is used to make the final assembly. Finally, the catadioptric camera system is calibrated using the omnidirectional camera calibration toolbox developed by Mei and Rives (Mei and Rives, 2007).



Figure 6.6: Catadioptric system design and assembly using a transparent extruded thermoplastic acrylic tube and Firewire 640x480 pixel color camera.

6.2.2. Fisheye cameras and the unify model

Fisheye cameras, is a dioptric systems consist of a fisheye lens (high FOV) on a conventional camera, which allows having omnidirectional images with properties similar to the ones obtained with catadioptric systems. Fisheye cameras do not have a single projection centre but a locus of projection centres called diacaustic. However, as proven by (Ying and Hu, 2004a), it is reasonable to assume that the small projection locus can be approximated by a single viewpoint, if the calibration accuracy under this assumption is satisfied by the requirement of applications. This assumptions permits that the unified model presented above, and the calibration methods for central catadioptric cameras can be directly extended to fisheye cameras, as proven by (Mei and Rives, 2007), and (Scaramuzza et al., 2006). For some of the applications developed in this thesis a commercial fisheye lens mounted on a digital conventional camera have been used. Figure 6.7, shows the camera on the Pelican UAV, a captured dioptric image, and the projection on the unitary sphere. The dioptric system has been calibrated using the proposed model and the open source toolbox developed by Mei (Mei and Rives, 2007).

6.3. UAV attitude estimation based on omnidirectional image processing

Omnidirectional vision can be a useful sensor for estimating attitude in UAV. It can be used as the only sensor or as complementary sensor for inertial and GPS information. Its main advantage is that the *Pitch* and *Roll* are estimated accordingly to the horizon line. Therefore it regards the land level variations. It also estimates *Yaw* according to visual objects that can be used for trajectory planning and See&Avoid strategies.

In general, there are twelve variables that define the UAV state (Beard,

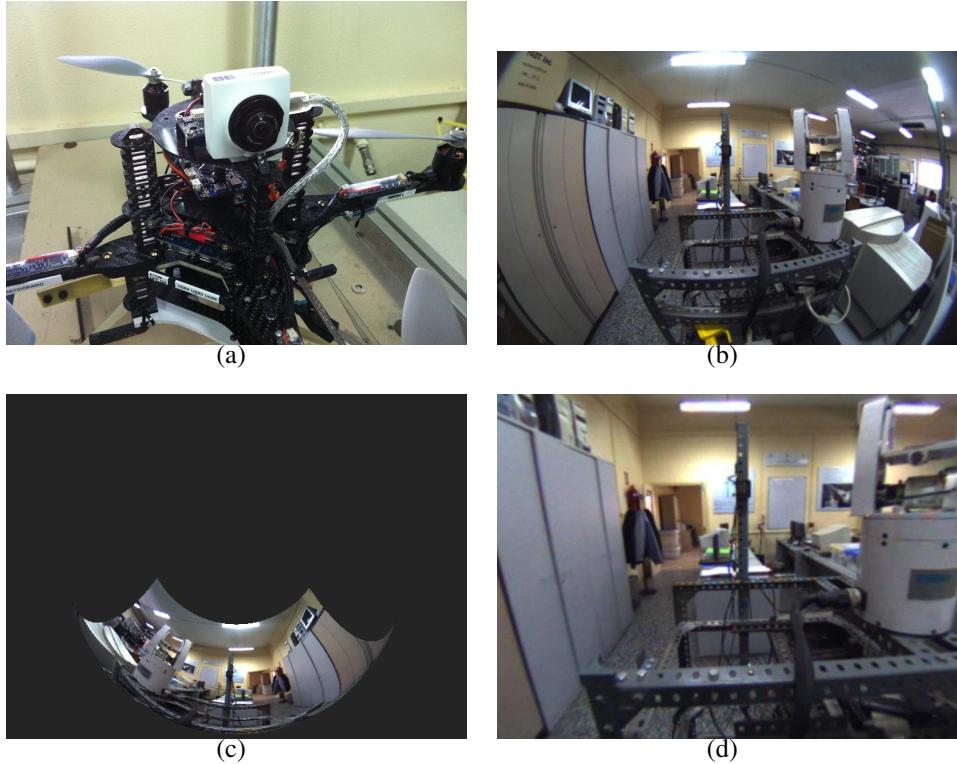


Figure 6.7: The unify model on Fisheye images: The fisheye lens in a conventional camera mounted on the Pelican UAV (6.7(a)), A dioptric image take with the fisheye image(6.7(b)), the obtained projection on the unitary sphere (6.7(c)) and a perspective image generated from the unitary model (6.7(d)).

2007), which are inertial latitude, longitude and altitude of the UAV, body frame velocities (rates); and *Roll*, *Pitch* and *Yaw* angles and rates. In order to make a low level controller for UAVs, it is necessary to have a direct measurement of the attitude angles (*Roll*, *Pitch* and *Yaw*) or their rates. Frequently, these variables are measured by using rate gyros as a part of a more complex sensor (the Inertial Measurement Unit (IMU)) that involves gyroscopes and accelerometers to estimate the relative position, velocity and acceleration of a vehicle in motion. There has been an active improvement in precision, cost and size of the sensor in the last years, from mechanical devices through optical systems to MEMs sensors. Gyroscopes and IMU units are extremely sensitive to measurements errors caused by drift, generating wrong estimation in orientation after a long operation period. It makes it necessary to reference them to an external measurement system like GPS framework (Siciliano and Khatib, 2008). Moreover, they still can be affected by structural and mechanical fatigue, vibrations, temperature changes, electric interferences and others (Walraven, 2003), (Dumai and Winkler, 1990) causing an erroneous data or a sensor failure. In addition, small and micro UAV sometimes have restrictions in cost and sensors payload capacity restricting the use of other sensors.

The idea of using visual information for UAV attitude estimation is not new. The first experiments attempted to obtain the skyline from images taken by a single perspective camera looking forward on the aircraft. The images are used for estimating the *Roll* angle with a horizontal reference (Ettinger et al., 2002), (Todorovic et al., 2003), (Cornall et al., 2006), (Dusha et al., 2007). These works differ in the way that they segment the sky and ground and in how they estimate the horizon line.

Todorovic *et al.* (Todorovic et al., 2003) drives the horizon detection problem as image segmentation and object recognition applying statistical appearance models based on both color and texture clues. They make a Bayesian segmentation based on a statistical framework employing Hidden Markov Tree on the appearance models. Cornall et al.(Cornall et al., 2006) uses a techniques focused on be implemented on a small micro-controller. The algorithm uses a simplistic method where the horizon is approximated as a threshold of the blue color plane, determining the optimal threshold by Otsuis method and Kmeans. This approximation gives goods results under clear sky conditions. Dusha *et al.* applies morphological image processing and the Hough transform to detect the horizon line and derived the *Roll* and *Pitch* angles. In addition the use of optical flow to obtain the bodyframe rates.

Omnidirectional vision has also been used for UAV control and attitude estimation. Hrabar (Hrabar and Sukhatme, 2003) use an omnidirectional system for sideways-looking sensing on an autonomous helicopter by applying image unwrapping. Demonceaux *et al.*(Demonceaux et al., 2006), use a similar approach to the one presented in this paper, showing the advantages of using omnidirectional rather than perspective images for attitude estimation. They detect the horizon line on the catadioptric image using a Markov Random Fields segmentation and then project it on the equivalent sphere projection model for a Catadioptric system, showing a good performance (without a ground truth validation), on an off-line processed video sequence.

Catadioptric systems have also been used for robot odometry and relative orientation on outdoor vehicles (Labrosse, 2006), (Scaramuzza and Siegwart, 2008) showing that it is possible to estimate relative orientation and position of a mobile robot using appearance panoramic images.

This section shows the application of Omnidirectional images processing in order to obtain the attitude information of an Unmanned Aerial Vehicle as a RAW sensor data. Then, this data can be filtering to obtain a soft measurement of the UAV state and introduce it to a flight controller (Beard, 2007).



Figure 6.8: Skyline projection on the unitary sphere model. Skyline is the occluding contour of the earth sphere surface, whose projection on the equivalent unitary sphere model through a plane that intersect it forms the depicted red circle.

6.3.1. Skyline and catadioptric Image.

To be able to measure the body frame attitude based on the catadioptric image, it is necessary to know, how the skyline is projected onto the unitary sphere and onto the catadioptric plane. Geyer and Daniilidis (Geyer and Daniilidis, 2001) demonstrated that a line on the space projects as a great circle on the unitary sphere and as a conic on the image plane. Later, Ying and Hu (Ying and Hu, 2004b) demonstrated that the occluding contour of a sphere in space is projected onto a circle on the unit sphere or onto a conic in the catadioptric image plane. Considering the skyline as the occluding contour on the earth sphere surface, it requires to look for a small circle in the unitary sphere model or for a conic or ellipse on the image plane as proposed by Demonceaux *et al.* (Demonceaux et al., 2006), (see Figure 6.8 for an illustration).

Because the original datum obtained is the image projection, the skyline detection focused on isolating the sky from the ground in this image and then estimating the best adjusted ellipse to the skyline. In order to isolate the sky from the ground, an approach based on the method employed by Cornall *et al.* (Cornall et al., 2006) is applied. In this method, the *RGB* components of each pixel are weighted using the function $f(RGB) = 3B^2/(R+G+B)$. This function has shown very good results for sky-ground segmentation under different light and cloud conditions. For each resulting grayscale image from function $f(RGB)$

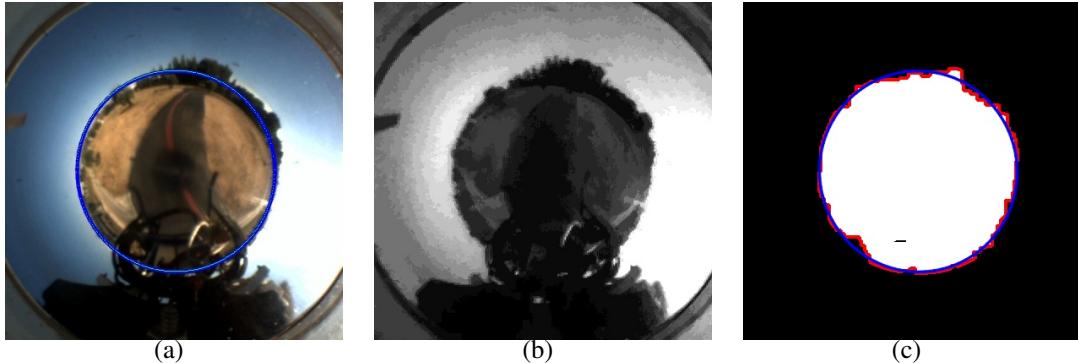


Figure 6.9: Skyline segmentation on a catadioptric image (6.9(a)), weighted using the function $f(RGB) = 3B^2/(R + G + B)$ (6.9(b)), and the obtained sky-ground binary image (6.9(c)).

a Pyramid Segmentation (Antonis, 1982) followed by a gaussian adaptive threshold function (Bradski and Kaehler, 2008) is used to obtain a sky-ground binary image. The pyramid segmentation reduces the effects of sunlight, brightness and cloud shadows in the image under variable lighting conditions. This threshold method is very fast, and produces good results in real-time. Figure 6.9 shows the segmentation on a catadioptric image weighted using the method described above.

Once a sky-ground thresholded image is obtained, the ground contour on the image can be easily defined. This contour represents the skyline and is used by a fitting function in order to obtain the ellipse with the best approximation to the contour. As it can be seen in Figure 6.9(c), if there are some high objects like mountains, buildings and trees that do not totally occlude the horizon, the segmentation method continues working because the approximate ellipse is obtained using all the segmented contour and not only parts of the skyline. Figure 6.10 shows some examples of the best fitted ellipse of the skyline on original catadioptrics images obtained during a UAV flights under different sunlight and weather conditions.

6.3.2. Skyline backprojection on sphere.

The segmented skyline is defined by the points of the contour that represent the ground border or by the adjusted ellipse points $\mathbf{Sky}_{img} = [x_{Sky_{img}}, y_{Sky_{img}}, 1]^T$. These points are projected back onto the unitary sphere using equation 6.5 obtaining $\mathbf{Sky}_s = [x_{Sky_s}, y_{Sky_s}, z_{Sky_s}]^T$ as shown in Figure 6.11. The circle formed by the skyline points on the sphere forms a plane that intersects with the unitary sphere. In order to obtain the parameters of the skyline

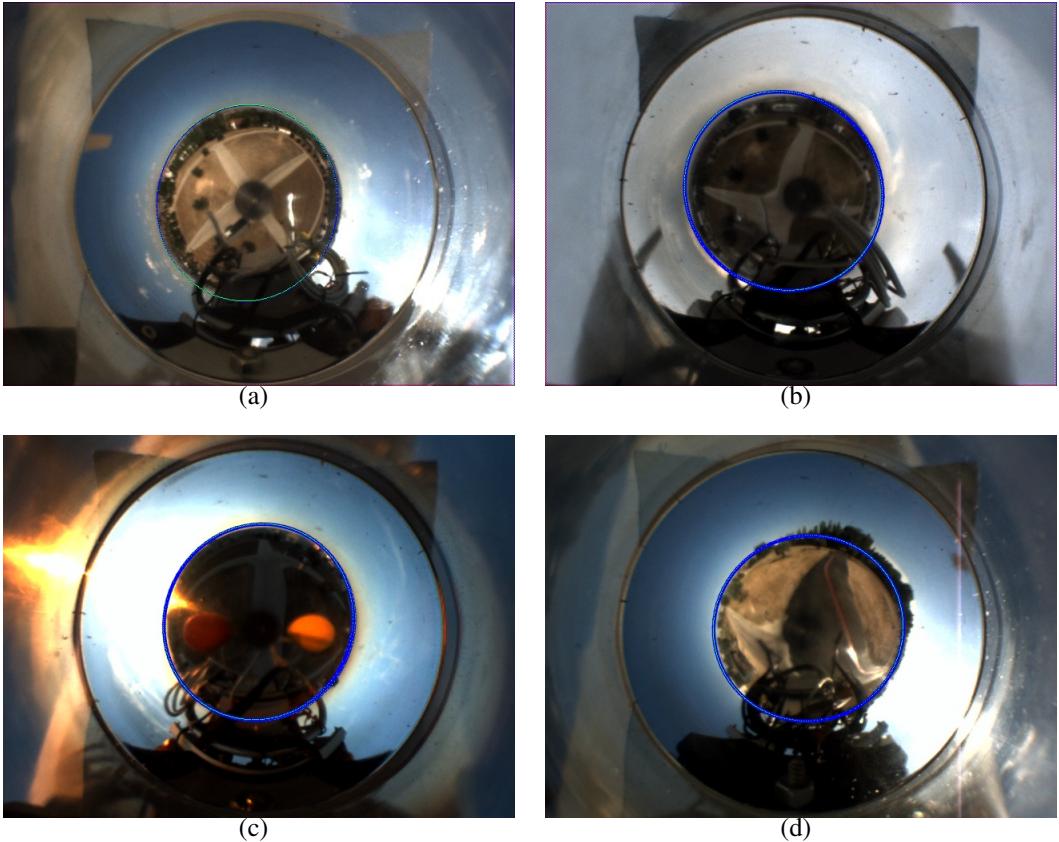


Figure 6.10: Examples of the best fitted ellipse of the skyline. The best fitted ellipse (blue) to the skyline on original catadioptric images obtained during a UAV flight under different sunlight and weather conditions. Figure 6.10(a) is a flight under a fall season clear sky, Figure 6.10(b) is a winter cloudy day, Figure 6.10(c) is on a winter partly cloudy sunset, and 6.10(d) is a takeoff and flight in a summer sunny day.

in the unitary sphere, it is sufficient to find the plane with normal equation $N_x x_{Sky_s} + N_y y_{Sky_s} + N_z z_{Sky_s} + D = 0$ that best adjusts the back-projected points of the skyline contour or the adjusted ellipse on the image plane.

For each point of the back-projected skyline, the normal equation of the plane is obtained by $z_{Sky_s}^i = N_x x_{Sky_s}^i + N_y y_{Sky_s}^i + D$ with $i = 1, \dots, n$ and an over-determined linear system of the form $(\mathbf{A}\mathbf{x} = \mathbf{b})$ is solved using the *pseudo-inverse method* to obtain the plane $\pi_{sky} = [N_x, N_y, 1, D]^T$ (Equation 6.15).

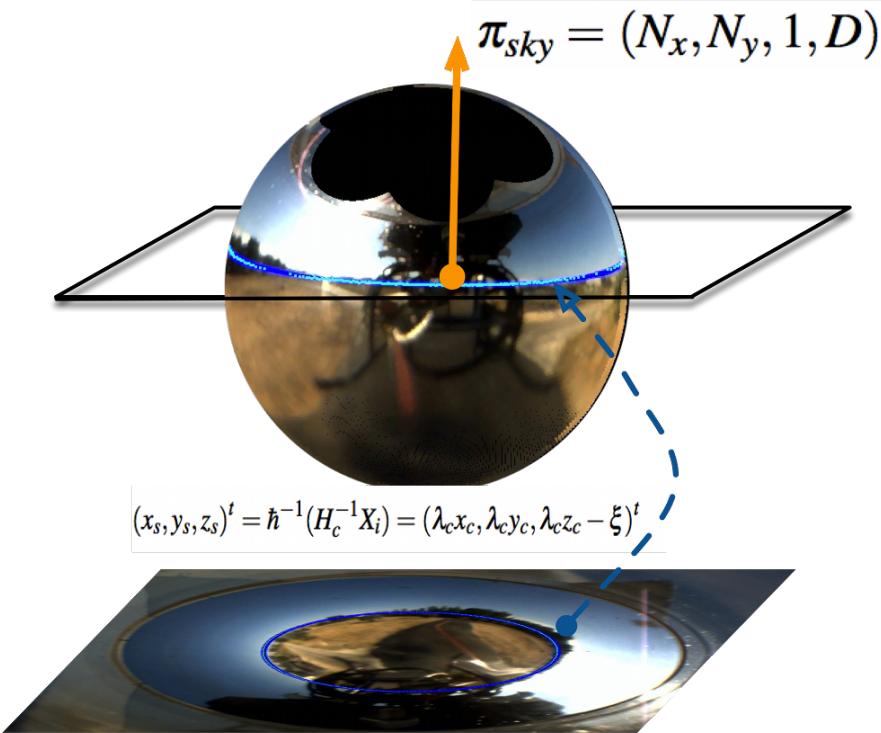


Figure 6.11: Attitude estimation based on the skyline projection on the unitary sphere. The best fitted ellipse (blue) to the skyline is back-projected onto a unitary sphere model, forming a plane that intersects the sphere (which forms a small circle). The normal vector to this plane defines the attitude of the camera and the UAV.

$$[N_x, N_y, D]^t = \arg \min_x \|\mathbf{Ax} - \mathbf{b}\|$$

where

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} x_{Sky_s}^i & y_{Sky_s}^i & 1 \\ \vdots & \vdots & \vdots \\ x_{Sky_s}^n & y_{Sky_s}^n & 1 \end{bmatrix} \\ \mathbf{x} &= \begin{bmatrix} N_x \\ N_y \\ D \end{bmatrix} \\ \mathbf{b} &= \begin{bmatrix} z_{Sky_s}^i \\ \vdots \\ z_{Sky_s}^n \end{bmatrix} \end{aligned} \tag{6.15}$$

6.3.3. Pitch and Roll estimation

The normal vector to the plane formed by the skyline and the unitary sphere is defined as $\mathbf{N} = [N_x, N_y, 1]^t$. Assuming that the camera frame is aligned with the UAV frame so that the x axis is the heading of the UAV and the y axis is aligned with the UAV wing, it is possible to obtain the desired *roll* (ϕ) and *pitch* (θ) angles, using equation 6.16.

$$\begin{aligned}\theta &= \arccos\left(\frac{N_x}{\sqrt{N_x^2 + N_y^2 + 1}}\right) \\ \phi &= \arccos\left(\frac{N_y}{\sqrt{N_x^2 + N_y^2 + 1}}\right)\end{aligned}\quad (6.16)$$

6.3.4. Yaw estimation using a visual compass

The relative heading of the UAV is calculated by using the so called visual compass on appearance images. This method was used by Labrosse (Labrosse, 2006) and later by Scaramuzza (Scaramuzza and Siegwart, 2008). It consists of a part of a panoramic image obtained from a catadioptric image using a polar to Cartesian coordinates change or unwrapping process employing equation 6.17

$$I(\alpha, R) = I_c(R \cos(\alpha) + u_0, R \sin(\alpha) + v_0) \quad (6.17)$$

where (u_0, v_0) are the coordinates of the catadioptric image center, α is a linear function with maximum range $[0, 2\pi]$ and R is a linear function that scans along the image Radius. The steps and range for α and R are defined according to the desired panoramic image resolution and size. The unwrapping process excludes the parts of the catadioptric images that involve the camera reflex and the outside of the mirror.

If the catadioptric image corresponds to a scene captured with an almost perfect vertical camera to the ground plane, then pure rotation will appear on the appearance image as a pure pixel column-wise shift. The relative rotation between two consecutive images is obtained, by finding the best match based on the images' column shift using the euclidean distance. Equation 6.18 shows the euclidean distance between two panoramic images I_m and I_n with the same size and space color, as a function of the column-wise shift on the Image I_n by α pixels (horizontal rotation). Figure 6.12 shows two consecutive appearance images obtained by an unwrapping process with a small rotation.

$$d(I_m, I_n, \alpha) = \sqrt{\sum_{i=1}^{width} \sum_{j=1}^{height} \sum_{k=1}^{Nchan} (I_m(i, j, k) - I_n(i + \alpha, j, k))^2} \quad (6.18)$$

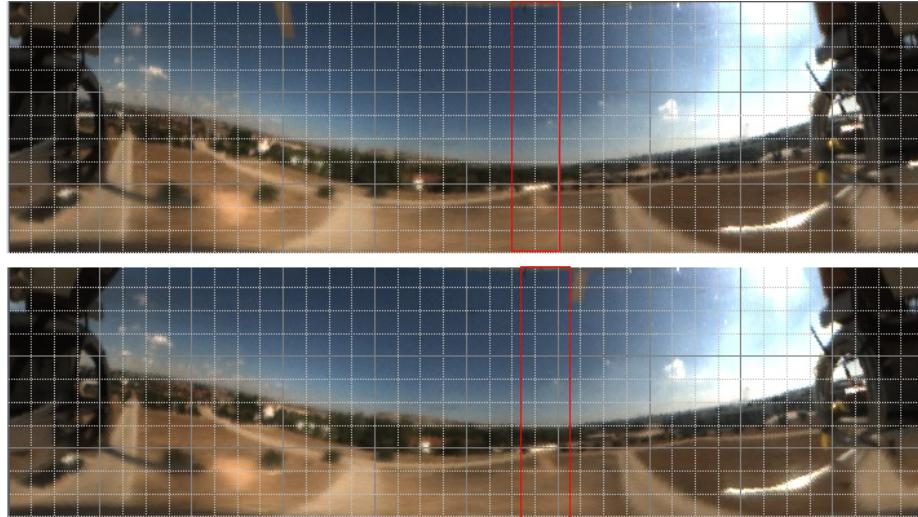


Figure 6.12: Two consecutive appearance images with a small rotation between them. A white grid is superimposed to reference and the red box shows clearly the column shift between images

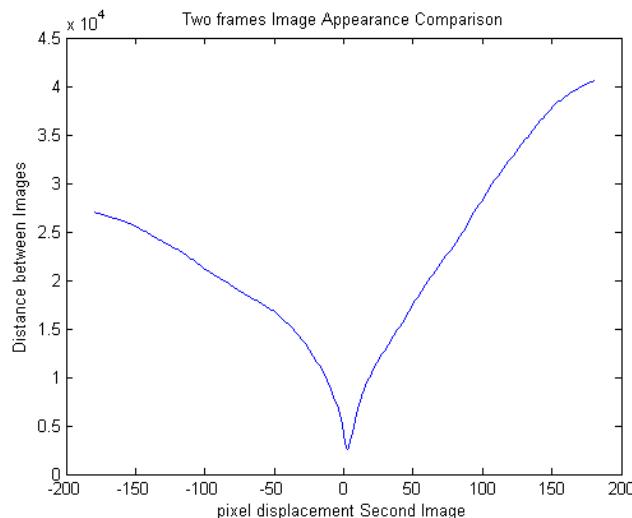


Figure 6.13: Euclidean distance between the two appearance images in Figure 6.12 as function of the column-wise shift on the second image.

The best shift α_{min} that minimize the distance function $d(I_m, I_n, \alpha_{min}) \leq d(I_m, I_n, \alpha) \forall \alpha \in \mathbb{R}$ is the best pixel rotation between these two images. Figure 6.13 shows the euclidean distance between the two appearance images in Figure 6.12 as a function of a column-wise shift on the second image.

The rotation angle or *Yaw* ψ between images is directly related to the obtained column shift between images, considering only the angular resolution of

the appearance images defined by the images field of view FOV and the images width as shown in equation 6.19.

$$\psi_{(I_m, I_n)} = \alpha_{min} \frac{FOV}{imgWidth} \quad (6.19)$$

In order to obtain the final rotation relative to the first image, it is necessary to add the obtained value to a counter.

The method described above was developed under the assumption of a pure camera rotation on its vertical axis which is perpendicular to the horizontal plane. In the general case, the UAV has translational components and *Roll* and *Pitch* variations, causing the camera vertical axis not to be at all times perpendicular to the horizon. However, as shown by Labrosse (Labrosse, 2006) and by Scaramuzza (Scaramuzza and Siegwart, 2008) the visual compass method based on appearance images is still valid under translation and attitude variations if the camera has small displacements or the distance to the objects is large compared with the displacement. Because images are captured at high frequency, small variations of *Pitch* and *Roll* are present between consecutive images; therefore, the pure rotation assumption is still valid. Finally, because the translational movements contribution to the optical flow is not homogeneous on the omnidirectional image, but the rotation movements contributes equally on all image, is enough to use the regions of the images on which the translation contribution to optical flow is reduced. Considering that a forward/backward translation has more contribution to the optical flow on the image regions corresponding to the sides of the vehicles and little in front and back regions of the vehicle, working only with the front or back portion of the image is enough to reduce the effects of a translation on the image rotation estimation. In our implementation a FOV of 120 degrees corresponding to the front central part of the panoramic image is used.

6.3.5. Tests and Results

Several tests have been made using the Colibri testbeds (CVG-UPM, 2010) in different seasons, weather and illumination conditions. Table 6.2 describes the flight and weather conditions then these tests have been done. The catadioptric system showed in Figure 6.6 have been employed for these tests. In these tests, a series of flights were performed in both autonomous and manual modes. In autonomous mode, the helicopter could take both, a previously defined trajectory, or a hovering (stationary) flight. In manual mode, a safety pilot takes a free flight with strong movements of the helicopter. The algorithm is tested during these flights (including a takeoff process) and an image sequence is stored, associating to each of the processed images the UAV attitude information estimated by the omnidirectional system. Moreover, a flightlog is created with the GPS position, the IMU data (heading, body frame angles and displacement velocities), and the helicopter position estimated by the Kalman Filter of the

Table 6.2: Flight test description and weather conditions.

Flight	Season	Weather & Lighting	Flight Mode
Flt. 1	autumn	Clear and windy day, 19°C	Auto Hover, high level
Flt. 2	winter	Cloudy Sky 2°C	Manual mode, high level
Flt. 3	winter	Partly Cloudy, Sunset 8°C	Manual mode, high level
Flt. 4	summer	Sunny Sky 37°C	Manual mode, takeoff-low level

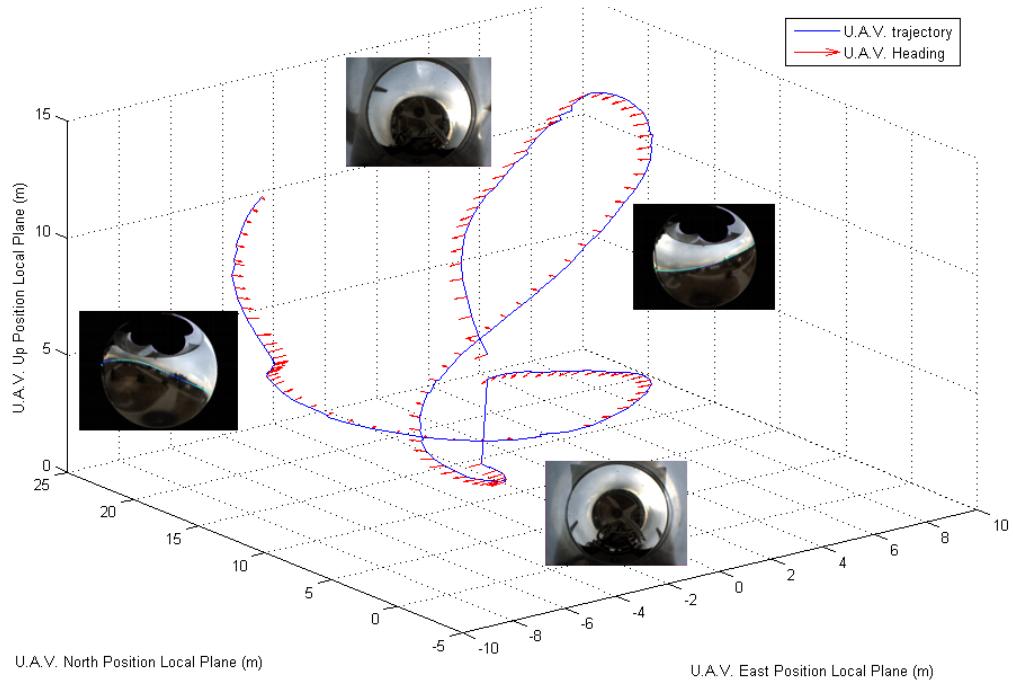


Figure 6.14: Flight number 2, 3D trajectory and UAV heading reconstruction, obtained using the flightlog data. The blue line depicts the translational movement and the red arrows represent the heading direction of the UAV (*Pitch* and *Yaw* angles). Superimposed images shows some of the different catadioptric and sphere result obtained during the flight sequence. In this flight, the algorithm takes a short break every five hundred frames (approximately 15 sec.) in order to store the acquired and processed images and IMU+GPS data in the on board hard disk, causing this strong change in the stored trajectory values.

controller on the local plane with reference to the takeoff point. These values are used for later comparisons with the estimated data using the catadioptric system. With catadioptric images of 640x480 pixels, the systems runs with an average of 20 fps on the Via Nano-ITX 1.5 GHz CPU on-board UAV.

Using the flightlog, it is possible to reconstruct the 3D trajectory of the vehi-

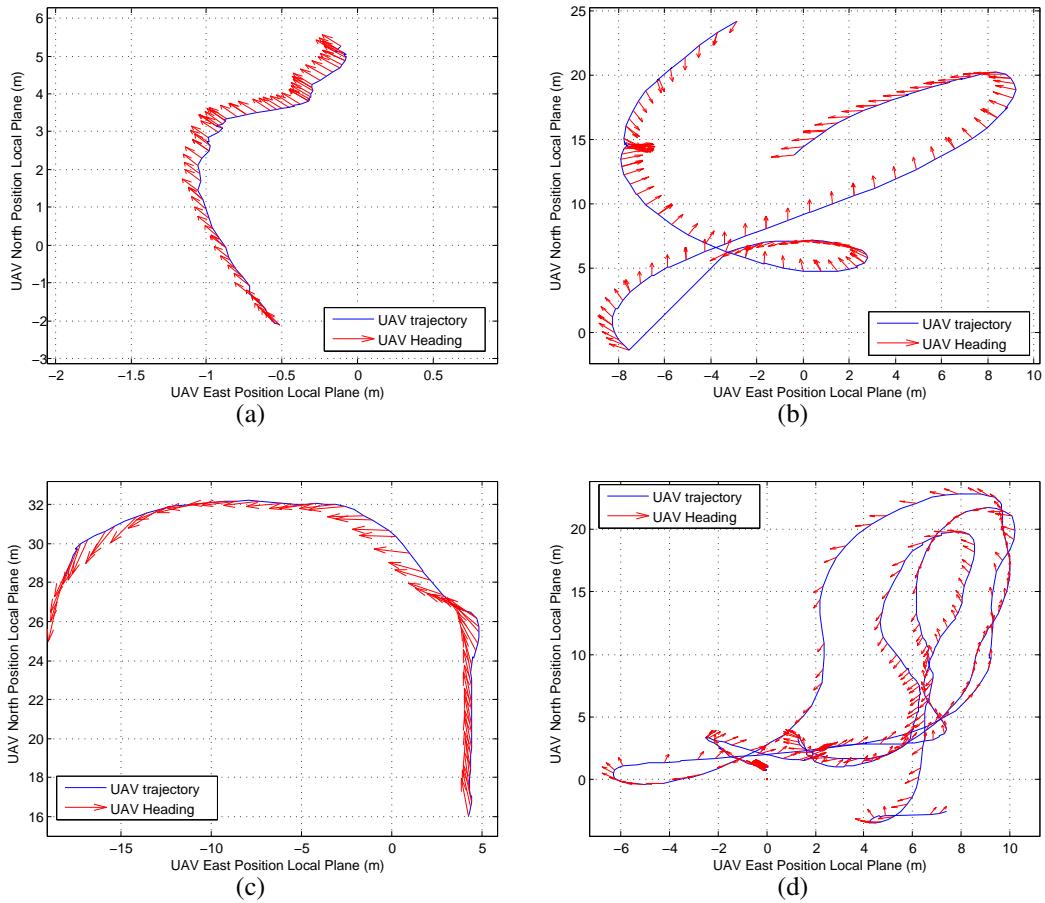


Figure 6.15: 2D trajectories and UAV heading reconstruction. Figure 6.15(a) is a high level hover flight in autonomous mode, Figure 6.15(b) is a manual high level flight, Figure 6.15(c) is manual high level during a sunset and 6.15(d) is a manual takeoff and low level flight.

cle, the camera and/or helicopter attitude. Figure 6.14 shows a 3D reconstruction of one flight made in manual mode, where it is possible to see big changes in the attitude and orientation of the UAV. Figure 6.15 shows the corresponding 2D reconstructions of flight tests described in table 6.2, showing for each one, the trajectory and helicopter heading.

The algorithm developed, estimates the absolute *Roll* and *Pitch* angles of the Camera bodyframe on the UAV and the relative *Yaw* rotation to the first image. For these tests, the camera is located on the pan and tilt platform of the helicopter in such a way that the vertical axes of the camera and helicopter are parallel (by adjusting the camera platform tilt). In this way, the hyperbolic mirror faces downward, and the camera looks up. This positioning ensures that all axes of the helicopter and camera are coincident, so that the obtained *Roll* and *Pitch* angles for the camera, are the same for the helicopter frame, as shown in Figure 6.8.

Table 6.3: RMSE for estimated *Roll* (ϕ), *Pitch* (θ) and *Relative Yaw* (ψ).

	<i>Roll</i> (ϕ)	<i>Pitch</i> (θ)	<i>R. Yaw</i> (ψ)
Flight 1	0.1807	4.3710	1.2533
Flight 2	2.7528	2.8267	10.7844
Flight 3	0.9363	4.0918	6.0413
Flight 4	2.8647	1.9183	5.0086

The estimated values of *Roll*, *Pitch* and *Yaw* from test flights (Table 6.2, Figure 6.15) are compared with the corresponding stored IMU values. Figures 6.16, 6.17 and 6.18 shows these results and table 6.3 has the Root Mean Squared Error RMSE of the estimated values compared with the IMU values as ground truth. In flight 2 (Figure 6.14), every five hundred frames (15 sec.), the algorithm made a small break to store the acquired and processed images and IMU data in the on-board hard disk (other tests, the process is continuous), causing this strong change in the measured values since not processed images are taken during this saving time.

The estimated *Roll* values have a very similar behavior to the one performed by the helicopter during the tests. The estimated values are very close to the IMU values and have a small RMSE against absolute values measured in the IMU.

Pitch values are also estimated accurately compared with the ground truth IMU. However when the helicopter has a high nose-up angle, a portion of the ground is occluded on the catadioptric image by the platform's structure and the UAV's reflections, causing a small error in the adjustment of the skyline on the equivalent sphere projection, and the *Pitch* estimation. Additionally, this causes that in general, the *Pitch* RMSE has higher values than the *Roll*, although these values are still a high-quality measurement. The error caused by the ground occlusion can be solved changing the camera position to be totally below the UAV bodyframe reducing the portion of the camera platform reflected on the mirror.

Yaw estimation uses the first image taken by the algorithm as a reference, calculating rotation with respect to this reference image. Absolute *Yaw* data measured by the IMU is rotated according to the first image angle and changed to a range between $0 < \psi < 360$ for easy comparison with omnidirectional data. Results show that the rotation between frames, as well as the total rotation, are both a good approximation to real values, however there are some cases

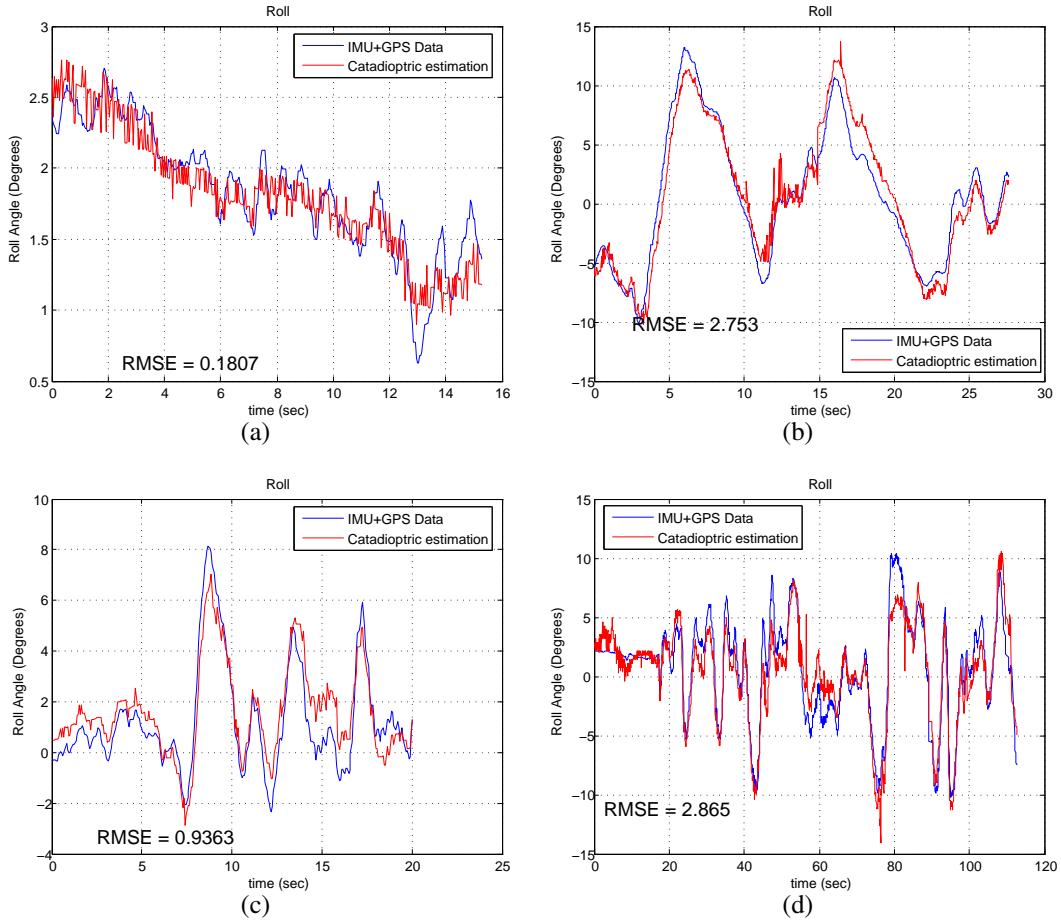


Figure 6.16: Estimated *Roll* compared with the *Roll* angle measured by IMU (*Roll* angle in degrees $-180 < \phi < 180$. Zero is level with the horizon and increasing is right wing down). Figures 6.16(a) to 6.16(d) corresponds to the flights shown in Figure 6.15.

in which the rotation direction is correctly estimated, but the magnitude is not correct, as can be seen in Figure 6.18(b). This happens, because *Yaw* estimation is obtained using a counter of rotations between consecutive images, causing an accumulative error along all the estimation process. In addition, the unwrapped panoramic image only has a resolution of 0.5 pixels per degree, causing that small rotations between consecutive images cannot be detected and not added to the counter. *Roll* and *Pitch* estimation do not have this accumulative error problem, because they are calculated by only using the information available in the last catadioptric image.

In Flight 4 (Figure 6.15(d)) the algorithm was running from the helicopter's takeoff phase in order to test the robustness of the skyline detection at low altitudes. During the first 20 seconds, the helicopter is on the ground, making a rotor speed up in order to reach the necessary revolutions (RPM) to lifting

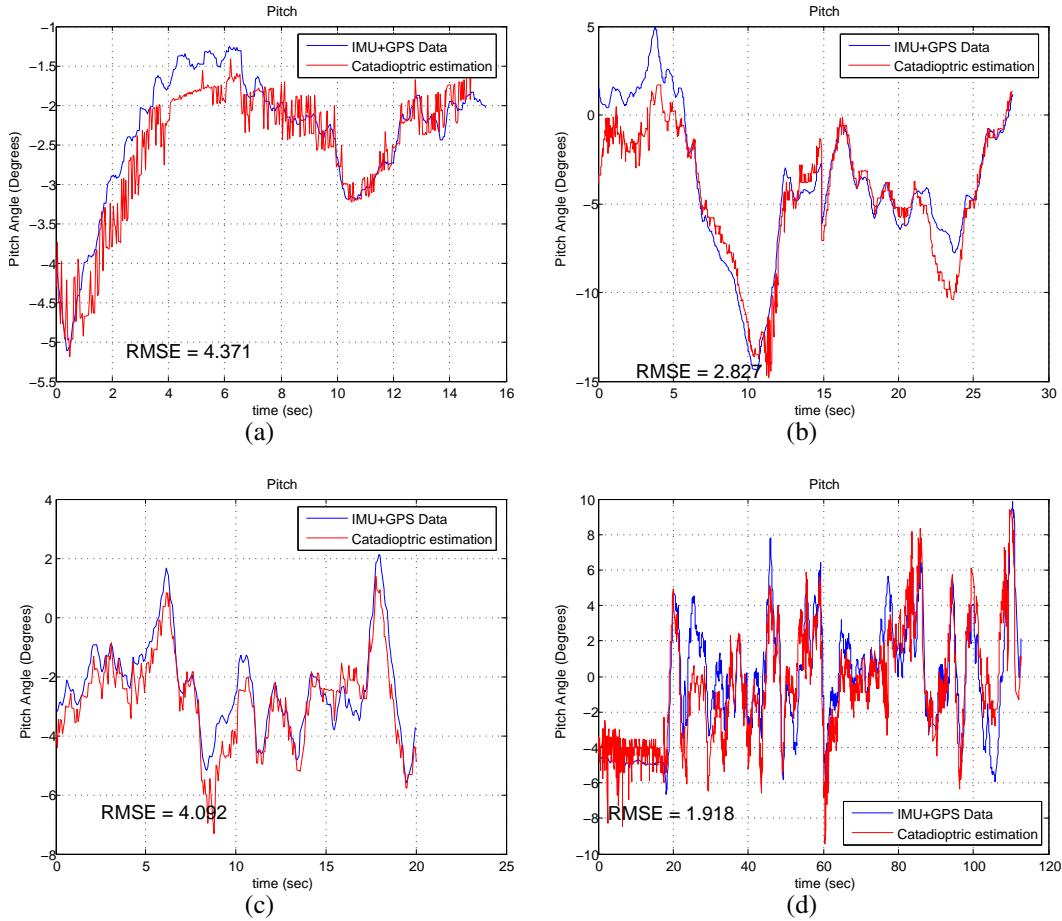


Figure 6.17: Estimated *Pitch* compared with the *Pitch* angle measured by IMU (*Pitch* angle in degrees $-90 < \theta < 90$. Zero is level with the horizon and increasing is nose up). Figures 6.17(a) to 6.17(d) corresponds to the flights shown in Figure 6.15.

the helicopter into the air, causing vibration on the camera platform. Skyline segmentation works fine on the captured images, but because the platform has a vibration component, the attitude angles have a big noise component on the estimated values during the beginning of this test as Figures 6.16(d), 6.17(d) and 6.18(d) show. This noise component during the takeoff produces a minimal increase of the RMSE values for this flight.

Tests have been done under different weather and sky conditions, flying at different heights and including images sequences taken with the helicopter on the ground and during a takeoff phase. In all cases, results are good, showing the feasibility of using a catadioptric system as a UAV attitude and heading estimator or as a redundant visual system. The total video sequences for these flights and additional tests in manual and autonomous mode are available on the Vision4UAV project Web Page (CVG-UPM, 2010)

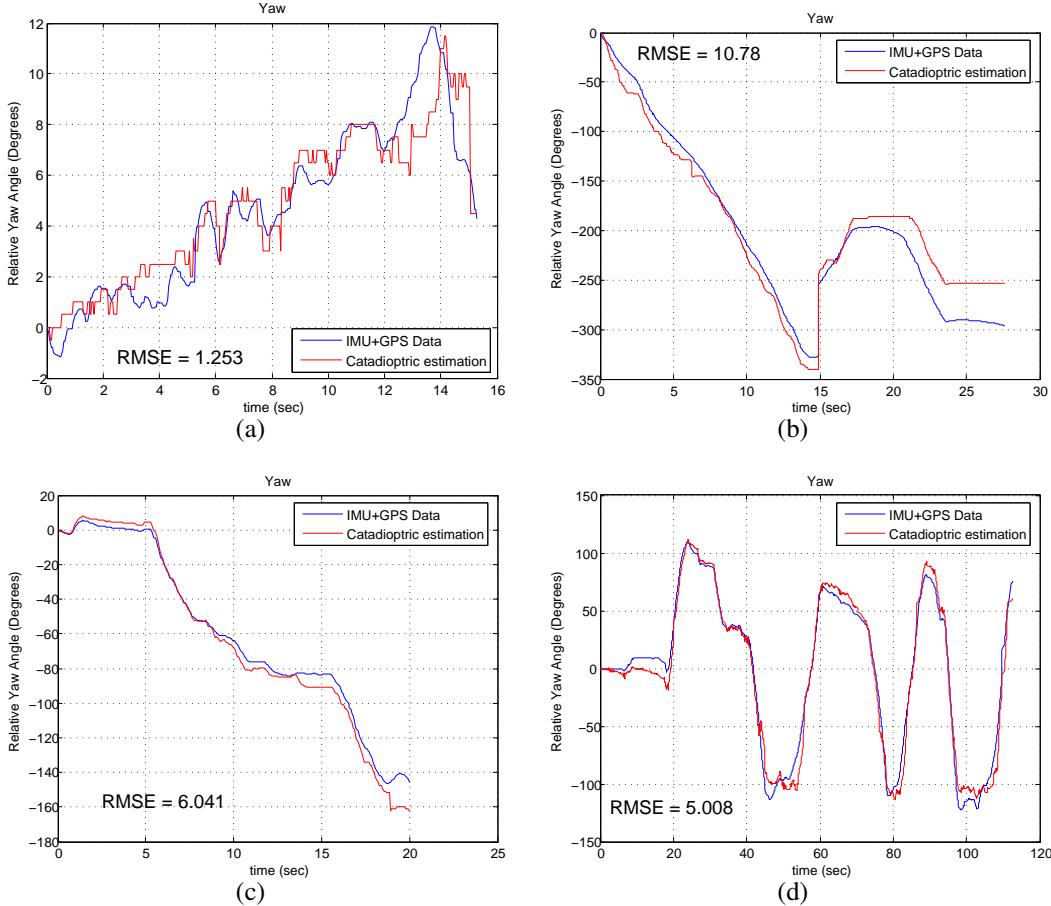


Figure 6.18: Estimated *Relative Yaw* compared with the *Relative Yaw* angle measured by IMU (Heading (*Yaw*) angle in Degrees. $-180 < \psi < 180$, zero is due north and increasing is rotating clockwise when viewed from above in absolute measuring and $-360 < \psi < 360$ in relative mode). Figures 6.18(a) to 6.18(d) corresponds to the flights shown in Figure 6.15.

6.4. UAVs See&Avoid

In this section, an application for collision avoidance based on visual information (See&Avoid) is presented¹. The method only uses visual data provided by an omnidirectional system. The developed method does not need for external information like the trajectory of the object with risk of collision or the distance to the object.

Considering two different vehicle moving in a horizontal plane with constant

¹This work has been done in collaboration with the Australian Research Centre for Aerospace Automation ARCAA (ARCAA et al., 2010) as part of two staff interchange stays under the International Cooperation Program for Unmanned Aerial Systems Research and Development ICPUAS (ICPUAS et al., 2009) on summer 2010 (visit to ARCAA) and winter 2010 (visit to CVG-UPM)

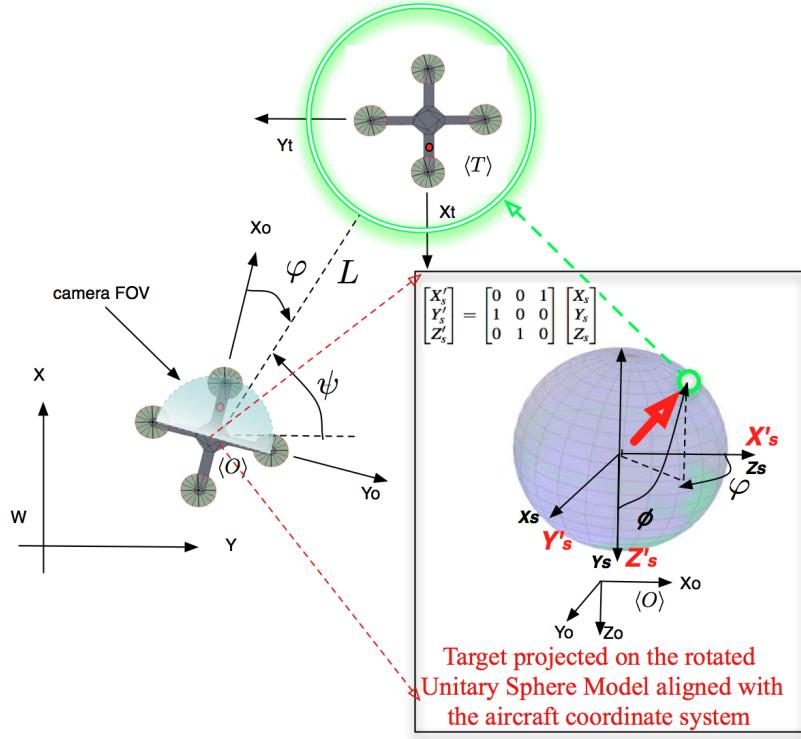


Figure 6.19: Owncraft-Target See&Avoid setup in 2D. The camera is modeled throughout the unitary sphere projection. Because the camera is looking forward and in order to align the sphere axes with the aircraft coordinate system, there is a change of coordinates of the poles. This also allows avoiding possible singularities when projected features are close to the pole.

altitude as shown in Figure 6.19, The kinematics of each vehicle is given by:

$$v_x = v \cos \psi, \quad v_y = v \sin \psi, \quad \omega_z = \omega \quad (6.20)$$

where $[x, y]^T \in \Re^2$ is the location of the vehicle and $\psi \in [-\pi, \pi]$ is the heading, both in world coordinates W . This model is constrained by the vehicle's maximum turning rate and linear velocity, i.e., $v_{min} < v < v_{max}$ and $|\omega| < \omega_{max}$, respectively. The two vehicles involved in the scenario are *own-craft* (O) (vehicle with camera) and *target* (T) (vehicle to avoid). The control inputs of (O) and (T) are linear and angular velocities $[\mathbf{v}_O \ \omega_O]^T$ and $[\mathbf{v}_T \ \omega_T]^T$, respectively. Where $\mathbf{v} = [v_x \ v_y \ v_z]^T$ and $\omega = [\omega_x \ \omega_y \ \omega_z]^T$.

Alternatively, the system can be modeled using polar coordinates where L is the euclidean distance between the own-craft and target and the angle φ is the view angle (or relative bearing) from the own-craft to the target. This angle is given by the on-board image processing based on the position of the target object on the omnidirectional image (described below). The omnidirectional image

is modeled using the unitary sphere projection model. Because the camera is looking forward and in order to align the sphere axes with the aircraft coordinate system, there is a change of coordinates of the poles. This coordinates change also allows avoiding possible singularities when projected features are close to the pole, as is explained in (Corke, 2010),

With regard to Figure 6.19, the whole system kinematics can be written as follows:

$$\dot{\mathbf{s}} = \mathbf{J}(\mathbf{s}, L) \mathbf{V}_O \quad (6.21)$$

where $\mathbf{s} = [\phi, \varphi]^T$ is the feature vector with ϕ and φ being the elevation and bearing of the target w.r.t. own-craft², $\mathbf{V}_O = [\mathbf{v}_O, \omega_O]^T$ as defined above and $\mathbf{J}(\mathbf{s}, L) = \mathbf{J}(\mathbf{s}, L)_{trans} + \mathbf{J}(\mathbf{s})_{rot}$ the Jacobian which can be decomposed in a translational and rotational part (see (Corke and Hutchinson, 2001), (Corke, 2010) for more details). Equation (6.21) can be used to define the control task following a similar approach to the one presented in Equation 5.6) such as

$$\mathbf{V}_O = -\lambda \mathbf{J}(\mathbf{s}, L)^+ \mathbf{e} \quad (6.22)$$

where $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ is the error signal, with $\lambda > 0$ (selected to have an exponential decouple decrease of the error $\dot{\mathbf{e}} + \lambda \mathbf{e} = 0$) and \mathbf{s}^* the desired feature value and $\mathbf{J}(\mathbf{s}, L)^+$ the Jacobian pseudo-inverse (Chaumette and Hutchinson, 2006; Chaumette and Hutchinson, 2007). The aim is to minimize the error signal by maximizing the desired value of our features ($\varphi \rightarrow \pm \frac{\pi}{2}$). This is achieved by using visual servoing approach (Hutchinson et al., 1996),(Corke, 2010). The details are presented in section 6.4.2.

Finally, it is proposed the hypothesis that collision avoidance is achieved when L is higher than twice the radius of the sphere that contains each vehicle ($L > 2R$). Furthermore, L is maximum when $|\varphi|$ is maximum. We guarantee a minimum separation between a vehicle and a target by maximising the view angle given by the omnidirectional sensor. This hypothesis is demonstrated on section 6.4.3

6.4.1. Detection Approach

Tracking approach

The object tracking was achieved by exploiting the color characteristic of the target and using a similar approach the one presented on section 5.4.1. The target is defined by a basic color (blue), attaching a simple colored mark to it

²note: it is assumed that camera and vehicle are aligned, approximately in the same point. The proposed control law gives the vector V_O of the camera modeled throughout the unitary sphere. Because the poles rotation on the unitary sphere, the vector V_O of velocities for the UAV corresponds to the one obtained from the control law defined on the rotated unitary sphere

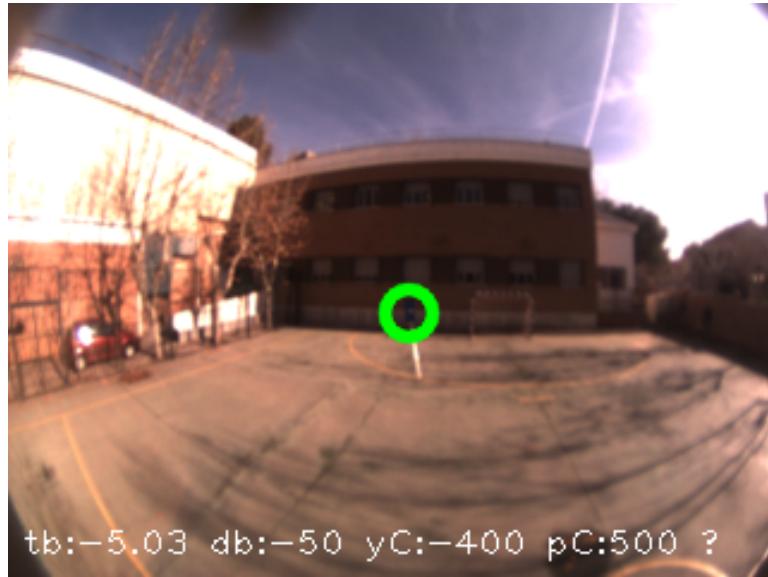


Figure 6.20: Angle estimation on the own-craft. Angle φ (tb in the image) is the relative bearing to the target, db is desired target bearing, yC is Yaw command and pC is Pitch command.

and tracking this mark. The Camshift color tracker is employed to continuously identify the selected object on the images sequences. This tracker has proven to be robust against apparent color variations caused by illumination changes, as well as large vibrations and partial occlusions. Figure 6.20, shows an example of the object tracked using the Camshift algorithm.

Target attitude estimation

For these tests, the Fisheye lens presented in Section 6.2.2 has been employed. The Camshift algorithm is used for tracking a target on the original images given by the Fisheye camera, obtaining at each iteration the image coordinates of the target point $\mathbf{X}_{T_c} = [x_{t_c}, y_{t_c}, 1]^T$. The coordinates of the target point on the image plane are back projected onto the unitary sphere model using equation 6.5, obtaining the target projection point on the unitary sphere $\mathbf{X}_{t_s} = [x_{t_s}, y_{t_s}, z_{t_s}]^T$. Because, the camera is looking forward, the camera optical axis z_s corresponds to the aircraft x axis, similarly, the x_s axis correspond to the aircraft y axis and the y_s axis corresponds to the aircraft z axis. As is explained in (Corke, 2010), the pole coordinates of the unitary sphere can be rotated in order to be aligned with a defined coordinate system or in order to avoid singularities caused by features projected closed to the pole. This coordinates changes, still allows designing a controller based on the projected features. So, the target projected features on the unitary sphere \mathbf{X}_{t_s} are rotated in order to be aligned with the aircraft coordinate systems using $\mathbf{X}'_{t_s} = [x'_{t_s}, y'_{t_s}, z'_{t_s}]^T = \mathbf{R}_{s2q} [x_{t_s}, y_{t_s}, z_{t_s}]^T$,

where \mathbf{R}_{s2q} is defined as equation 6.23.

$$\mathbf{R}_{s2q} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (6.23)$$

Finally, (6.24) is employed to obtain the target attitude with respect the UAV. This equation already includes the rotation between the camera unitary sphere and the aircraft coordinate systems.

$$\phi_t = \arctan \left(\frac{\sqrt{x'_{ts}^2 + y'_{ts}^2}}{z'_{ts}} \right); \varphi_t = \arctan \left(\frac{y'_{ts}}{x'_{ts}} \right) \quad (6.24)$$

where ϕ_t and φ_t represent the target elevation and bearing angles w.r.t the rotated unitary sphere, aligned with the aircraft coordinate system.

6.4.2. Approach Implementation

See&Avoid control is achieved by using a two stage control architecture. A low level controller is responsible for attitude stabilization. It is embedded, closed, unmodifiable but gains are tunable B.2. This low-level controller accepts desired attitude values (*Roll*, *Pitch*, *Yaw*) that then converts internally to motor commands. Coupled to this low-level controller is a vision-based controller, which sends desired attitude values based on image features. Therefore, a visual servoing approach (Hutchinson et al., 1996) is used and defined as follows.

Assuming that the two angles in (6.24) are obtained from the detection phase and following a similar approach to that used on section 5.2.3, it is possible to define on the unitary sphere surface, the feature vector $\mathbf{s} = [\phi, \varphi]^T$ and the desired position vector $\mathbf{s}^* = [\phi^*, \varphi^*]^T$. The time derivative of the target position ($\mathbf{X}_{To} = [X_{To}, Y_{To}, Z_{To}]^T$) w.r.t own-craft is defined as $\dot{\mathbf{X}}_{To} = -\mathbf{v}_O - \boldsymbol{\omega}_O \times \mathbf{X}_{To}$ and is similar to the one defined in equation 5.3.

Considering that $X_{To} = L \sin \phi \cos \varphi$, $Y_{To} = L \sin \phi \sin \varphi$ and $Z_{To} = L \cos \phi$, and integrating the time derivatives of equation 6.24 in Equations 5.3) it is possible to obtain the Jacobian that relates the measured features on the unitary sphere with camera movements on the tridimensional space, as proven by Corke (Corke, 2010).

$$\mathbf{J}(\mathbf{s}, L) = \begin{bmatrix} \frac{\cos \phi \cos \varphi}{L} & \frac{\cos \phi \sin \varphi}{L} & \frac{\sin \phi}{L} & -\frac{\sin(\varphi)}{\sin \phi} & -\frac{\cos(\varphi)}{\sin \phi} & 0 \\ -\frac{L \sin \varphi}{L \sin \phi} & \frac{L \cos \varphi}{L \sin \phi} & 0 & -\frac{\cos \phi \cos \varphi}{\sin \phi} & -\frac{\cos \phi \sin \varphi}{\sin \phi} & 1 \end{bmatrix} \quad (6.25)$$

Equation 6.25 can be re-writing in a two component form

$$\dot{\mathbf{s}} = \mathbf{J}(\mathbf{s}, L) \mathbf{v}_O = \{\mathbf{J}(\mathbf{s}, L)_{trans} + \mathbf{J}(\mathbf{s})_{rot}\} \mathbf{v}_O \quad (6.26)$$

where each component of $\mathbf{J}(\mathbf{s}, L)$ is a 2×3 matrix. The translational part is a function of depth and location of the image features, however the rotational only depends on location. Thus, (6.26) can be re-written as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \frac{1}{L} \mathbf{J}(\mathbf{s})_{trans} [\mathbf{v}_O]^T + \mathbf{J}(\mathbf{s})_{rot} [\boldsymbol{\omega}_O]^T \quad (6.27)$$

This equation fully relates velocity of features in the image plane with camera (vehicle) linear and angular velocities. The dependency in L and invertibility of \mathbf{J} imposes some constrains in the minimum number of features, type of servoing (Hutchinson et al., 1996) and knowledge of feature depth. To deal with these constrains the following assumptions are made:

1. Vehicle altitude is kept constant during the experiments. This leads us to approximate $v_{z_O} = 0$
2. Velocity in x is constant ($v_x = 1, \forall t \neq 0$)
3. Velocity in y is set to zero initially, i.e, $v_y = 0, t = 0$ however $v_y \neq 0, \forall t > 0$
4. Vehicle *Roll* and *Pitch* are approximately constant, therefore angular velocities in x and y are negligible. $\omega_{x_O} \approx \omega_{y_O} \approx 0$
5. Target distance is large enough to consider features at infinity ($\sin \varphi \approx 0$) and vehicles are almost on the same horizontal plane ($\sin \phi \approx 1$). This lead us to a direct relationship between optical flow and angular velocity. This is possible because of the constant elements 0 and 1 in the jacobian (equation 6.25), indicating that elevation is invariant to rotation about the Z axis and relative bearing angle is invariant to Z axis translation.

Taking the vector containing $\dot{\phi}$ in (6.27), and rewriting

$$\dot{\phi} = \frac{1}{L} [J(s)_{trans21} \quad J(s)_{trans22} \quad 0] \begin{bmatrix} v_{x_O} \\ v_{y_O} \\ v_{z_O} \end{bmatrix} + [J(s)_{rot21} \quad J(s)_{rot22} \quad 1] \begin{bmatrix} \omega_{x_O} \\ \omega_{y_O} \\ \omega_{z_O} \end{bmatrix} \quad (6.28)$$

Finally, applying the proposed assumptions in Equation 6.28, the control task defined on 6.22 can be expressed as equation 6.29 shows:

$$\omega_{z_O} = -\lambda(\varphi_t - \varphi_t^*) \quad (6.29)$$

This control law is used for keeping the target at a given angle $\varphi_t > 0$ in the image plane while the vehicle moves with constant velocity. λ is selected in such a way that scale the rotational velocity commands generated by the control law in order to be compatible with the input velocity commands received by the low level controller is explained below.

6.4.3. Tests and results using quadrotors

Experimental tests were performed using a Pelican quadrotor B.2 and a static target (flight trial complexity is greatly reduce if a static target is used initially). The testbed has a low-level stability controller based on PID that uses information from GPS, IMU, pressure altimeter and magnetometer fused using a Kalman filter.

On-board vision processing is achieved using a Dual core Atom 1.6 GHz processor with 1 GB RAM. The vision-based controller (running in the Atom) communicates throughout a serial interface with the low-level autopilot. This controller sends velocity references to the autopilot creating a *dynamic look-and-move* servoing architecture as the one showed in Figure 5.16. The algorithm runs with a frequency of 20 fps on the on-board Atom Board, that allows getting a faster response from the low lever controller to the generated collision evasion commands.

A series of flights were performed for these test. Pelican quadrotor was commanded to make a constant movement along X axis with a velocity of 0.5 m/sec at a fixed altitude. The visual algorithm was used in order to track a desired target (a blue mark in the space) and when the collision target was correctly detected and tracked, the evasion commands generated by the visual algorithm were sent to the low level controller. Figure 6.21 shows the collision avoidance trajectory and aircraft heading executed by the quadrotor once the collision target was detected for four different test. Vehicle evasive trajectory depends on initial target detection in the image. It will move left around the target if $\varphi_0 > 0$ setting $\varphi_t^* = 50^\circ$, and right around the target if $\varphi_0 < 0$ setting $\varphi_t^* = -50^\circ$.

When using monocular sensors and no range to target is available, the issue of *when to break the avoidance maneuver* is still an open problem. Therefore, proposals such as constraining own-craft heading, dynamic change of desired view angle, thresholding line of sight rate, passive ranging are yet still to be tested in this context and benefits assessed. Because it is not implemented an avoidance ending system, the quadrotor continues maintaining the target at the

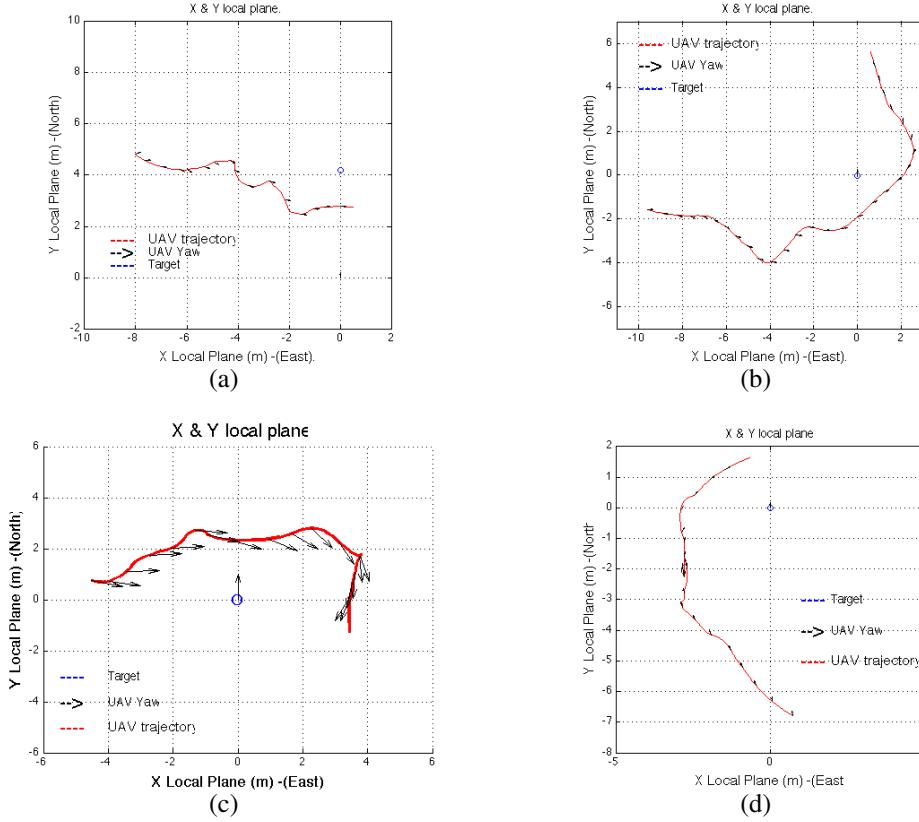


Figure 6.21: 2D trajectories and UAV heading reconstruction for See&Avoid Test. Figures 6.21(a) and 6.21(b) corresponds to two different test with object evasion by the right side. Figures 6.21(c) and 6.21(d) corresponds to two different test with object evasion by the left side.

given angle $\varphi_t > 0$ until the maneuver is manually finished.

Figure 6.22 shows the distance between the own-craft and the target and the minimum distance obtained, while Figure 6.23 shows the target position angle on the image plane, the generated command and the aircraft position for the four test presented in Figure 6.21. It is clear that the experimental result validates the proposed hypothesis *maximizing avoidance angle will maximize minimum avoidance distance*.

Test have validated the proposed approach for See&Avoid based only on visual information given by a omnidirectional sensor, warranting that maximizing the avoidance angle will maximize the minimum avoidance distance to the target.

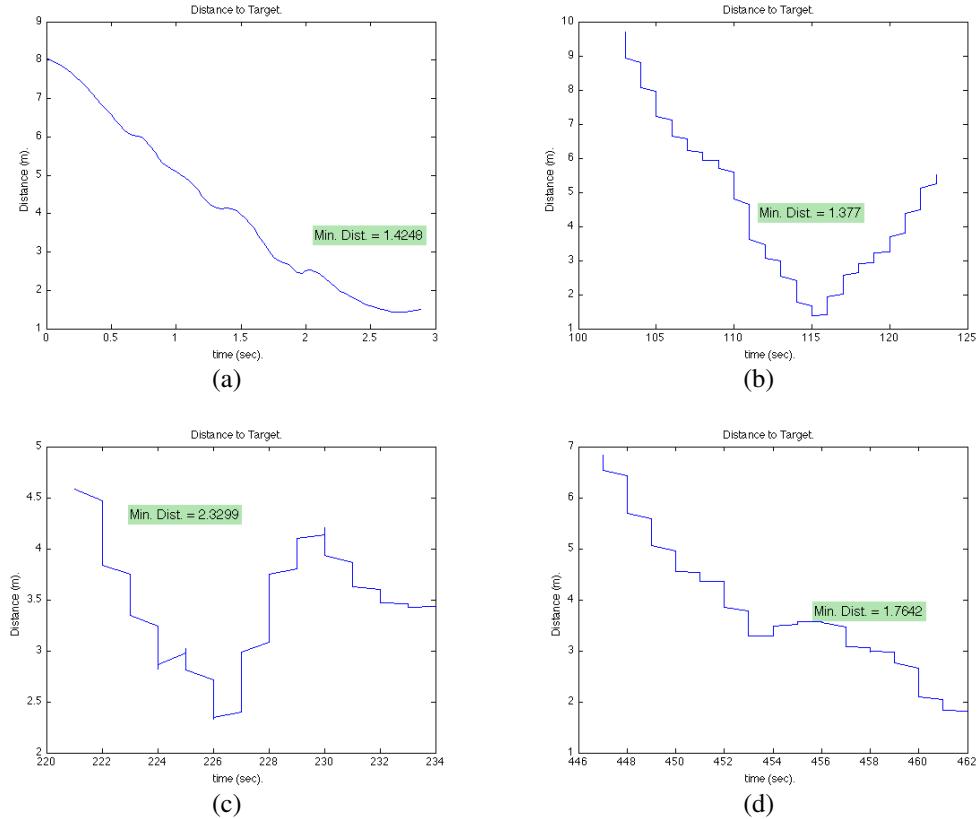


Figure 6.22: UAV to Target Distance for See&Avoid Test. Figures 6.22(a) to 6.22(d) corresponds to the flights shown in Figure 6.21.

6.5. Conclusions

This chapter deals with the research and results using omnidirectional computer vision techniques on-board a UAV for attitude estimation and visual control.

A method for UAV attitude (*Roll* and *Pitch*) and heading estimation based totally on visual information taken by a catadioptric camera have been developed and successfully tested. This approach has been validated against inertial measures using a UAV testbed, showing that the estimated values are very good approximation of the real state of the aircraft, demonstrating the feasibility of using this kind of system as a main sensor on UAVs with reduced sensor payloads or as a redundant system for IMU and gyroscopes in cases of failure or malfunction of these systems. The *Pitch* and *Roll* are calculated in relation to the horizon and they are thereafter referenced to the actual land inclination, which can vary during the flight. Similarly, the heading is calculated in reference to external objects, so it can be used with advantage in trajectory planning.

The algorithm has been tested under different weather conditions, showing

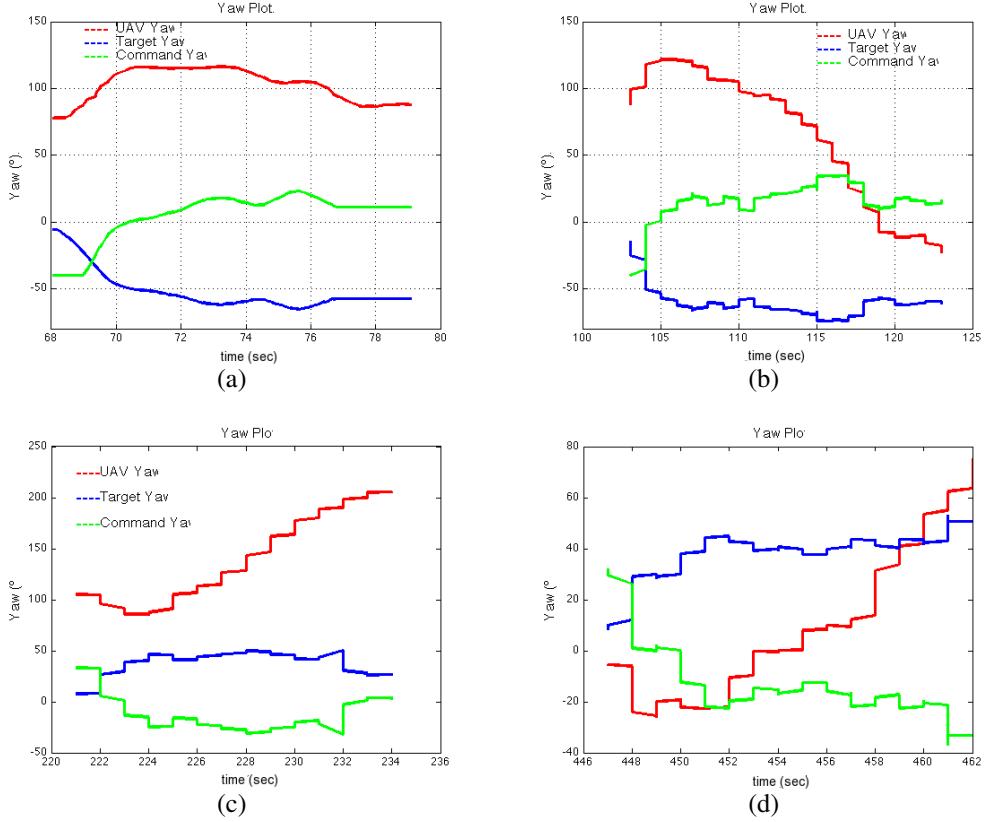


Figure 6.23: Estimated relative target Yaw compared with the *Relative Yaw* angle measured by IMU (Heading (*Yaw*) angle in degrees and the command Yaw generated. $-180 < \psi < 180$, zero is due north and increasing is rotating clockwise when viewed from above in absolute measuring and $-360 < \psi < 360$ in relative mode.). Figures 6.23(a) to 6.23(d) corresponds to the flights shown in Figure 6.21.

the adaptability of the proposed method to much sunlight and illumination, and the robustness in extreme cases such as cloudy skies or a sunset, which make skyline isolation difficult to obtain.

The tests have shown a good performance at different flight levels, with a high robustness at low altitudes where high objects (trees, mountains and buildings) often occlude parts of the contour of the earth sphere surface, making it difficult to detect and segment the skyline. At high altitudes, these objects are under the horizon line level, allowing an excellent sky ground segmentation and algorithm efficiency.

Estimated values using the omnidirectional system, have shown a good response to strong changes on the aircraft flying state with a near real time computational cost. These visual based measurements have proven to vary

according to the measurements of the classical sensors, showing discrepancies of only a few degrees. These discrepancies can be used for sensor fusion, as well as for obstacle avoidance and trajectory updating. Considering these facts, the inertial data measured with the catadioptric system are appropriate for a flight controller based on visual sensor with additional features like object tracking and servoing.

This chapter has also presented a robust real time method for UAVs See&Avoid based only on the visual information provided by an omnidirectional camera sensor. The system was tested on real UAV flights with collision scenarios, showing that the proposed method is an efficient technique for real time evasion. In addition, the proposed hypothesis that *maximizing avoidance angle will maximize minimum avoidance distance* has been demonstrated with experimental results.

The algorithm for Sense&Avoid exploits the properties given by fisheye lenses and catadioptric cameras calibrated using the unify model for central cameras, in order to detect, track and evade a collision target by maximizing its angle on the image plane: This method allows fast response on the evasion maneuver, guaranteeing a successful collision avoidance with a minimum distance, which is twice higher than the radius of the sphere that contains the vehicles ($L > 2R$).

Chapter 7

Conclusions and Future Work

In this dissertation, a series of contributions to the field of visual control for UAVs have been presented. The proposed methods varies from image enhancement, state estimation (pose and attitude) up to visual control applications like object following, visual tracking, visual positioning and landing, and See&Avoid methods. The proposed approach has been experimentally demonstrated showing the feasibility and capabilities of using visual systems for the control of UAVs.

7.1. Conclusions

Computer vision on-board UAVs has demonstrated being a useful tool for improving the capabilities of aerial vehicle. It is therefore a technology that is not only limited to being “eye on the sky”. In this thesis, it has been demonstrated that a single camera can be exploited in order to improve the state estimation and become a decisive sensor in carrying task such as tracking, navigation and positioning. It is also a principal instrument for dynamic environments where traditional sensors have a limited capacity.

The main contributions of this thesis is to use the basic concepts of computer vision and control in order to increase the autonomy of UAV by means of the following research threads: visual processing techniques for image tracking,

3D pose and attitude estimation based on image information, generic visual applications for aerial vehicles, and visual servoing for UAV.

The principal contributions derived from this thesis are summarized as follows;

■ **Tridimensional pose estimation based on visual tracking:**

A UAV 3D pose estimation method using planar object tracking has been developed and validated. The method exploits the rich information obtained by a projective transformation of planar objects on a calibrated camera. The algorithm obtains the metric and projective components (relative position and attitude) of a reference object (landmark or helipad) with respect to the UAV camera coordinate system, using robust object tracking method based on homographies and feature matching.

■ **Control systems using a decoupled architecture:**

This thesis has proposed and validated different methods in order to develop high level decoupled visual servoing architectures that directly relates controlled variables with unique feature input. The decoupled control methods developed using this methodology, have been tested on real aerial test, showing the robustness to drive the non-holonomic behavior of rotary wing vehicles and the perturbations commonly presented on outdoors flights as well as demonstrating a large versatility on the on-board integration and tuning process.

■ **UAV control based on visual 3D pose estimation:**

Aircraft control based on the position and attitude of the UAV with respect a landmark in order to make accurately positioning, hovering and landing and closing a high level control loop using a *dynamic position base look-and-move* architecture.

■ **UAV control based on visual features:**

The classic approaches for visual servoing are extended to cover diverse kind of cameras (pinhole and unified omnidirectional) generating control structures that allows defining practical control systems like aerial moving objects following and See&Avoid systems for light UAV.

■ **Omnidirectional visual systems and UAV applications:**

The unified theory for central catadioptric cameras is used for modeling large FOV dioptric and catadioptric cameras designed to be used on-board light UAV. These modeled cameras, have been employed to increase the FOV of the aircraft without increasing the complexity of the visual system. Benefit of these systems has been demonstrate by means of two contributions: the first contribution uses a calibrated catadioptric camera for an on-board attitude and heading estimation. It shows how the skyline

projection in the catadioptric image is segmented and used for calculating the UAV's attitude, and how the appearance images obtained are used in order to obtain a visual compass. It is a useful tool to calculating the relative rotation and heading of the aerial vehicle. The second contribution uses a calibrated fisheye lens in order to design novel See&Avoid applications based on object tracking. Both approaches have been experimentally validated.

■ **UAV video enhancement :**

The general projective model for perspective images and the unified theory for omnidirectional images are exploited to make general proposed algorithms for UAV, such as mosaic building, video stabilization, panoramas and spherical image construction, visual tracking among others. These techniques have been implemented to be used on-board light UAVs with different series of visual systems, like conventional perspective cameras, eye fish lenses and catadioptric systems.

Image enhancement techniques have been tested in a variety of UAVs, ranging from light UAVs up to general aviation aircraft, showing a good performance for image mosaicing and real time video stabilization in all of them. Otherwise, visual servoing techniques have only been tested in rotary wing light UAV, showing the advantages of using visual information in a high level hierarchical control system to improve the capabilities of this kind of VTOL vehicles.

Currently, visual systems are employed as an aid for robotic control and navigation. However, recent developments in parallel processing and the large variety of visual sensors are moving the design of control systems in order to directly integrate a visual sensor on its main process. This thesis has demonstrated that it is currently possible to design and build a control system for UAV that employs visual information as a main sensor with accuracy, performance, reliability similar to traditional sensors normally employed in UAV control. These properties are complemented, with the proven advantage of being a "eye in the sky" for human operators.

Different state of the art algorithms have been implemented for being used on-board UAV applications, looking for a balance between speed, robustness and reliability. They give enough performance for applications in UAV where there is a high rate of noise, vibration, illumination and change in movement, among other. Nevertheless, these algorithms have already been implemented on-board in reduced size and low weight CPUs. There is a great thread looking for the best optimization through the use of parallel processing units that allows increasing the performance and the amount of data that current algorithms are capable to process. In this thesis, one of the methods developed has been partially integrated in GPU, that allows increasing the processing time on images

with a double resolution up to 5 times faster than on a standard CPU unit.

This thesis has demonstrated that several visual processing techniques, which are novel, low cost and with low weight image sensors can be used for estimating and measuring most of the parameters involved in the flight state and control of a UAV. It also includes attitude, altitude and pose estimation, in addition to external dynamic information of the aircraft surroundings, that cannot often be measured with other inexpensive instruments. Test and results have shown that estimated data based on processed image can be compared with information provided by traditional sensors, and be a useful source of information suitable for being used as a complementary sensor for a low cost and limited payload UAS.

7.2. Future Work

The methods and applications proposed and validated on this thesis have demonstrated that visual information is a good source of information that can be integrated in control systems for UAV. In order to achieve this goal, there are different areas that must be investigated, from which the following points have been considered fundamental:

- Parallel computing has demonstrated the capability of processing large quantity of information and making repetitive processing in a concurrent way, that is often optimal and faster than sequential processing, which is currently used by most visual systems. Parallel computing is now possible to be used in on-board UAV due to constant developments in GPU cards. They have allowed the integration of multiple cores with high processing capability in smaller, light and low power consumption processing units. Considering this constant increase in its processing capabilities, current image processing algorithms can be employed in higher resolution images with a higher processing frequency, solving many of the current limitations that are present in sequential systems.
- UAVs have complex dynamics that demand processing algorithms to be robust to high changes in perspective, illumination or high presence of noise. The study, development and optimization of general image processing algorithms that allow identifying and tracking points, corners, features and regions (blobs) under changing conditions are a primordial area for visual processing with applications for general robotics and UAV.
- The growing interest in using UAV for civilian and industrial fields will demand a series of visual systems and image processing technique that

can adapt to different situations, task and applications like inspection, surveillance, law enforcement, among other. These conditions requires visual systems and flight control systems that can be easily adapt to these situations without involving a significant increase in the control system, payload or cost. The current developments of visual systems and its integrations on UAV control systems have to be designed in a way that they can easily adapt to a variety of applications and operation conditions. In such manner, an open source framework and integration protocol have become a fundamental key factor in order to permit the integration of diverse visual components and systems with the large range flight controllers and aerial robots systems in the current and future market..

Appendix A

Publications derived from this thesis

A.1. Journals with JCR

Below is a list that summarizes the Journal derived from the works presented on this thesis. Related information about the ranking given for the category *Robotics* by the Journal Citation Reports (JCR) (ThomsonReuters, 2010) for the year 2010 is provided.

On-board and Ground Visual Pose Estimation Techniques for UAV Control: Carol Martínez, Iván F. Mondragón, Pascual Campoy, Miguel A. Olivares, *Journal of Intelligent and Robotic Systems*, ISSN: 0921-0296 (Print) 1573-0409 (Online). Volume 61, Numbers 1-4, 301-320, DOI: 10.1007/s10846-010-9505-9, 2011. **JCR: 0.757, last third.**

Unmanned aerial vehicles UAVs attitude, height, motion estimation and control using visual systems: Iván F. Mondragón, Pascual Campoy, Carol Martínez, Miguel A. Olivares, Luís Mejías, *Journal of Autonomous Robots*, Volume 29, Number 1, 17-34, DOI: 10.1007/s10514-010-9183-2, 2010. **JCR: 2.011, first third.**

Omnidirectional vision applied to unmanned aerial vehicles (UAVs) attitude and heading estimation: Iván F. Mondragón, Pascual Campoy, Carol

Martinez, Miguel A. Olivares, Robotics and Autonomous Systems, Volume 58, Issue 6, Omnidirectional Robot Vision, 30 June 2010, Pages 809-819, ISSN 0921-8890, DOI: 10.1016/j.robot.2010.02.012. **JCR: 1.313, second third.**

Visual 3-D SLAM from UAVs: Jorge Artieda, Jose Maria Sebastian, Pascual Campoy, Juan Correa, Iván Mondragón, Carol Martinez, Miguel Olivares. Journal of Intelligent and Robotic Systems, ISSN: 0921-0296 (Print) 1573-0409 (Online). DOI 10.1007/s10846-008-9304-8 January 2009. **JCR: 0.757, last third.**

Computer Vision Onboard UAVs for civilian tasks: Pascual Campoy, Juan Correa, Iván Mondragón, Carol Martinez, Miguel Olivares, Jorge Artieda, Luís Mejías. Journal of Intelligent and Robotic Systems, ISSN: 0921-0296 (Print) 1573-0409 (Online). DOI 10.1007/s10846-008-9256-z. August 2008. **JCR: 0.757, last third .**

A.2. Congress

Omnidirectional bearing-only see-and-avoid for small aerial robots: Iván F. Mondragón, Luís Mejías, Pascual Campoy. Accepted for Proceedings of the 5th IEEE International Conference on Automation, Robotics and Applications ICARA 2011. Wellington, New Zealand. Dec. 6-8, 2011.

3D Object following based on visual information for Unmanned Aerial Vehicles: Iván F. Mondragón, Pascual Campoy, Carol Martinez, Miguel A. Olivares. Accepted for Proceedings of IEEE Latin American Robotics Competition (LARC) and The Latin American Robotics Symposium (LARS) LARC & LARS & CCAC 2011. Bogotá, Colombia. Oct. 1-4, 2011.

Non-symmetric membership function for fuzzy-based visual servoing on-board a UAV: Miguel A. Olivares, Pascual Campoy, Iván F. Mondragón, Carol Martinez. 2010 international FLINS conference on foundations and applications of computational intelligence (FLINS 2010). Chengdu, China. August 2-4, 2010.

Fuzzy controller for UAV-landing task using 3d-position visual estimation: Miguel A. Olivares, Iván F. Mondragón, Pascual Campoy, Carol Martinez. proceedings of IEEE world congress on computational intelligence (IEEE WCCI 2010-IEEEFUZZ2010). Barcelona, Spain, july 18-23, 2010.

3D pose estimation based on planar object tracking for UAVs control: Iván F. Mondragón, Pascual Campoy, Carol Martinez, Miguel A. Olivares. Proceedings of IEEE international conference on robotics and automation

ICRA2010. Anchorage, Alaska, USA. May 3-8, 2010.

Onboard and ground visual pose estimation techniques for UAV control:
Pascual Campoy, Iván F. Mondragón, Carol Martinez,, Miguel A. Olivares. 3rd international symposium on unmanned aerial vehicles (UAV'10). Dubai, Arab Emirate, june 21-23, 2010.

A pan-tilt camera fuzzy vision controller on an unmanned aerial vehicle:
Miguel A. Olivares, Iván F. Mondragón, Pascual Campoy, Carol Martinez. 2009 IEEE/RSJ international conference on intelligent robots and systems (IROS), St Louis, Usa. Oct 11-15,2009,

Trinocular ground system to control UAVs: Carol Martinez, Iván F. Mondragón, Pascual Campoy, Miguel A. Olivares. 2009 IEEE/RSJ international conference on intelligent robots and systems (IROS), St Louis, Usa. Oct 11-15,2009,

Visual servoing using fuzzy controllers on an unmanned aerial vehicle:
Miguel A. Olivares, Pascual Campoy, Iván F. Mondragón, Carol Martinez. EUROFUSE 2009. workshop on on preference modelling and decision analysis.

Vision for guidance and control of UAVs in civilian tasks: Pascual Campoy, Iván F. Mondragón, Carol Martinez, Miguel A. Olivares. UAV'08 International Symposium On Unmanned Aerial Vehicles Orlando, Florida, USA. June 23-24, 2008.

Fuzzy Control System Navigation using Priority Areas: Miguel A. Olivares, Pascual Campoy, Iván F. Mondragón, Carol Martinez. 8th International FLINS Conference on Computational Intelligence in Decision and Control. Madrid, Spain . September 2008.

Visual Model Feature Tracking for UAV Control: Iván F. Mondragón, Pascual Campoy, Juan Correa, Luís Mejías. Proceedings IEEE Internacional Symposium on Intelligent Signal Processing WISP2007. Alcalá de Henares, Spain. October 3-5 de 2007.

Stereo Visual System for Autonomous Air Vehicle Navigation: Luís Mejías, Pascual Campoy, Iván F. Mondragón, Patrick Doherty. IFAC Symposium on Intelligent Autonomous Vehicles. IAV 2007. Proceedings of 6th IFAC Symposium on Intelligent Autonomous Vehicles. IAV 2007. Toulouse, France,September 3-5, 2007

A.3. Book Chapters

Non-symmetric membership function for Fuzzy-based visual servoing on-board a UAV: Miguel A. Olivares, Pascual Campoy, Iván F. Mondragón, Carol Martínez. Computation Intelligence Foundations and Applications ISBN: 978-981-4324-69-4 981-4324-69-8, pages 300-307.

Visual Servoing for UAVs: Pascual Campoy, Iván F. Mondragón, Carol MartínezUAV'08, Miguel A. Olivares. Visual Servoing. ISBN 978-953-307-095-7, Publisher: InTech Publication date: April 2010.

Fuzzy control system navigation using priority areas: Miguel A. Olivares, Pascual Campoy, Iván F. Mondragón, Carol Martínez, Juan Correa. Computational Intelligence in Decision and Control. ISBN: 978-981-279-947-0(ebook) 981-279-947-8(ebook), pages 987-996

A.4. Digital media

Digital media related with this thesis is available at the vision4uav web page project (CVG-UPM, 2010) on the section dedicated to this thesis
<http://www.vision4uav.com/?q=phdthesis>

Appendix B

UAV specifications

In this appendix, there is a brief description of the UAVs employed on the development of this thesis. This description includes some aircrafts specifications, the visual system employed and the communications structures used for information exchange between the aircraft low level controller and the high level on-board visual system.

B.1. Gas and Electric powered UAV helicopters

The vision4uav project (CVG-UPM, 2010) has two Rotomotion UAV vehicles (Figure B.1), The first one is a modified Bergen industrial Twim RC helicopter SRA1. It is a gas powered helicopter. The second one is a modified Xcell Electric RC electric helicopter SR20. Table B.1 summarizes the principal technical characteristics.

B.1.1. Helicopter Autopilot API

External process (ground clients, and on-board system) interact with the Low Level AFCS autopilot through a UDP socket over and ethernet network. The API consist of a series of messages, that are used to send and get information

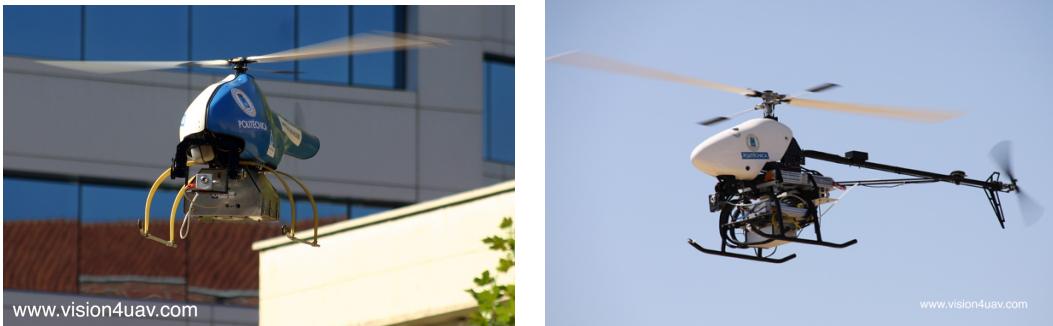


Figure B.1: UAVs rotary wing helicopters used on this thesis: (a) Rotomotion SRA1 Modified Bergen Industrial Twim gas powered.(b) Rotomotion SR20 modified Xcell Electric RC helicopter

from the autopilot. Following are summarized the most used messages.

Aircraft State: The state object encapsulates most all of the state of the helicopter and the autopilot system. It is sent at the highest rate of all the messages since it is the most useful for the ground station:

```
typedef struct {
    /* Euler angles relative to the ground */
    double phi;
    double theta;
    double psi;
    /* Body frame rotational rates */
    double p;
    double q;
    double r;
    /* Position relative to the ground */
    double x;
    double y;
    double z;
    /* Velocity over the ground */
    double vx;
    double vy;
    double vz;
    /* Raw magnetometer readings */
    double mx;
    double my;
    double mz;
    /* Body frame linear accelerations */
    double ax;
    double ay;
```

Table B.1: Rotomotion UAVs technical specifications

	SRA1 Gas powered	SR20 Electric powered
Length (mm)	1700	1700
Height (mm)	750	650
Main Blade Size (mm)	880	880
Tail Blade Size (mm)	130	130
Power Unit	Gasoline Motor @52 HP	Electric motor @ 1400 W.
Max Payload (kg)	14	4
Endurance (min)	45	15
Autopilot	AFCS 2.5	AFCS 3.4
Sensor	GPS, IMU and Magnetometer	GPS, IMU and Magnetometer
Communications	WiFi/Ethernet	WiFi/Ethernet
Classification	Light UAV	Light UAV
Visual System		
CPU	VIA mini-itx 1.0 GHz	VIA nano-itx 1.5 GHz
RAM (GB)	1.0	2.0
cameras interfaces	USB 2.0, 1394a, ethernet	USB 2.0, 1394a, ethernet, Analog captured
OS	linux	linux
Communication with Autopilot and Ground	WiFi/Ethernet	WiFi/Ethernet

```

        double          az;
/* raw body frame rotational rates */
        double          raw_p;
        double          raw_q;
        double          raw_r;
/* Miscellaneous */
        double          trace;
        double          voltage;
        uint16_t        resets;
        uint16_t        rpm0;
        uint16_t        rpm1;
        uint16_t        rpm2;
} state_t;

```

GPS state: The GPS state messages encode information about the quality of the position solution and the raw Lat/Lon coordinates for applications that are not satisfied with solely the local tangent plane. This allows geo-referencing of the sensor data and payloads, should it be required. The raw velocity and position values are run through a Kalman filter that uses the vdop and pdop to weight the readings. It uses the integrated velocity as a position state estimate, then aides that with the position measurements. This allows a higher quality of solution for

the local tangent plane.

```
typedef struct {
    int16_t      numsv;
    int16_t      sacc;
    int16_t      pacc;
    int16_t      fixtype;
    double       vel_n;
    double       vel_e;
    double       vel_d;
    double       pos_n;
    double       pos_e;
    double       pos_d;
    double       lat (raw_pos_n AFCS 2.5);
    double       long (raw_pos_e AFCS 2.5);
    double       alt (raw_pos_d AFCS 2.5);
    double       trace;
} msg_gps_t;
```

Position Control commands: It tells the AFCS where the operator wants the UAV to go

```
typedef struct {
    int16_t      pos_mode;
    int16_t      hdg_mode;
    int16_t      transit_time;
    uint16_t     pad1;
    double       x;
    double       y;
    double       z;
    double       hdg;
} msg_flyto_t;
```

Velocity teleoperation commands: Teleop messages are sent by custom ground stations with their own joysticks or controllers, or by ground based software that is handling targeting and/or navigation control. The velocities are roughly in m/s, but depend on the actual steady state offsets. Very small values may actually cause the aircraft to move backwards.

```
typedef struct {
    double       fore;
    double       side;
    double       alt;
    double       hdg;
} msg_teleop_t;
```

Helicopter desired position and heading confirmation: The desired position message is sent as a response to a flyto packet. It indicates the position in the NED frame to which the UAV will try to fly

```
typedef struct {
    double n;
    double e;
    double d;
    double hdg;
} msg_desired_pos_t;
```

Pan and Tilt platform control: If the UAV is outfitted with a camera platform, it will respond to this message. Only the fields relevant to the selected mode need to be filled in by the ground station; the AFCS will fill in all the fields in its reply.

```
typedef struct {
    int16_t mode;
    int16_t pad0;
    int16_t pad1;
    int16_t pad2;
    double relative_roll;
    double relative_pitch;
    double relative_yaw;
    double absolute_roll;
    double absolute_pitch;
    double absolute_yaw;
    double camera_n;
    double camera_e;
    double camera_d
} msg_desired_pos_t;
```

B.2. UAV quadcopter

The vision4uav project (CVG-UPM, 2010) has one AscTec Pelican Quadcopter (Figure B.2. Table B.2 summarizes the principal technical characteristics.

B.2.1. Quadcopter autopilot SDK

External process (ground clients, and on-board system) interact with the Low Level AscTec autopilot through a serial interface (running at 57600bps) and a series of messages, that are used to send and get information from the autopilot. Following are summarized the most used messages.

```
struct LL_STATUS{
// battery voltages in mV
    short battery_voltage_1;
```



Figure B.2: UAVs rotary wing quadcopter used on this thesis: Ascending Technologies Pelican UAV

```

    short          battery_voltage_2 ;
    short          status ;
    short          cpu_load ;
    char           compass_enabled ;
    char           chksum_error ;
    char           flying ;
    char           motors_on ;
    short          flightMode ;
    short          up_time ;

};

struct IMU_RAWDATA{
    int            pressure ;
    short          gyro_x ;
    short          gyro_y ;
    short          gyro_z ;
    short          mag_x ;
    short          mag_y ;
    short          mag_z ;
    short          acc_x ;
    short          acc_y ;
    short          acc_z ;
    unsigned short temp_gyro ;
    unsigned int   temp_ADC ;
};

struct IMU_CALCDATA{
    int            angle_nick ;
    int            angle_roll ;
    int            angle_yaw ;
    int            angvel_nick ;
};

```

Table B.2: Pelican AscTec UAV technical specifications

	AscTec Pelican
rotors	4
Area (mm ²)	600
Blade Size (mm)	250
Power Unit	4 Electric motor @120 W.
Max Payload (kg)	0.5
Endurance (min)	15
Autopilot	AscTec AutoPilot
Sensor	GPS, IMU, Magnetometer, and pressure altimeter
Communications	Serial Xbee RF
Classification	Light UAV
Visual System	
CPU	AscTec 1.6 GHz Atom Board
RAM (GB)	1.0
cameras interfaces	USB 2.0
OS	linux
Communication with Autopilot and Ground	Serial Interface /Wifi-Ethernet

```

int           angvel_roll;
int           angvel_yaw;
short         acc_x_calib;
short         acc_y_calib;
short         acc_z_calib;
short         acc_x;
short         acc_y;
short         acc_z;
int           acc_angle_nick;
int           acc_angle_roll;
int           acc_absolute_value;
int           Hx;
int           Hy;
int           Hz;
int           mag_heading;
int           speed_x;
int           speed_y;
int           speed_z;
int           height;
int           dheight;
int           dheight_reference;
int           height_reference;

```

```
};

struct GPS_DATA_ADVANCED{
    int             latitude;
    int             longitude;
    int             height;
    int             speed_x;
    int             speed_y;
    int             heading;
    unsigned int   horizontal_accuracy;
    unsigned int   vertical_accuracy;
    unsigned int   speed_accuracy;
    unsigned int   numSV;
    int             status;
    int             latitude_best_estimate;
    int             longitude_best_estimate;
    int             speed_x_best_estimate;
    int             speed_y_best_estimate;
};

struct CTRL_INPUT{
    short           pitch;
    short           roll;
    short           yaw;
    short           thrust;
    short           ctrl;
    short           checksum;
};

struct CTRL_INPUT CTRL_Input;
```

C

Appendix

Research Groups on UAS and Vision Systems

Several teams from MIT, Stanford, Berkeley, ARCAA and USC, UPM among others, have had an ongoing UAV project. They are multidisciplinary groups working on different areas related with UAS, some of them having an actively research on computer vision for aerial robotics. Below is a review of some of the most important institutions and their UAV projects. Finally, a compressive list of UAV manufacturers and research groups around the world, can be find on <http://uav.eas.gatech.edu>.

■ Universidad Politécnica de Madrid UPM - Spain

1. Computer Vision Group CVG - Universidad Politécnica de Madrid (UPM)

<http://vision4uav.com>

The main goal of the CVG in the field of aerial robotics, is the design and build of Unmanned Aerial Vehicles (UAVs) guided and controlled by computer visual systems for applications in civilian spaces. Research includes vision techniques for pose and attitude estimation,

visual enhancement, visual control and servoing among others.

2. (Robotics and Cybernetics Group - Universidad Politécnica de Madrid (UPM) -

<http://www.robcib.etsii.upm.es/>

The Robotics and Cybernetics research group worked on several topics within the fields of robotics, flexible manufacturing and biomedical engineering. In the field of UAV, they are currently focused on UAV system modeling and low-level control to mission planning, supervision and collision avoidance, going from vehicle constraints to mission constraints as well as Micro Aerial Vehicles indoor/outdoor navigation.

■ **Universitat Politècnica de Catalunya UPC - Spain**

ICARUS

<http://icarus.upc.edu>

The research areas includes fields like: Intelligent Communications and Avionics for Robust Unmanned Aerial Systems or Evaluation and new strategies for the smooth integration of civil mission oriented UAV in non-segregated airspace. ICARUS research project focus on the development of new hardware and software architectures to control the mission and payload in Unmanned Aerial Vehicles (UAVs).

■ **Centro Avanzado de Tecnologías Aeroespaciales CATEC - Spain.**

<http://www.catec.aero>

CATEC works on several topics on the industrial aviation sector, including R&D and technology transfer on the field of Avionics and ob-board systems, UAVs platforms, Automation and Robotics among others. On the field of Aerial Robotics have a a fleet of UAVs platforms used for systems integration and testing.

■ **Technische Universität Berlin - Germany**

Laboratory for autonomous flying robots

<http://pdv.cs.tu-berlin.de/lfafr/index.html>

They are working on different subjects connected with practical applications of autonomous aerial robots with a certain level of on-board intelligence. The main research areas include:mathematical modeling of small scale aerial robots, control of small scale aerial robots, sensors and sensor data processing for autonomous navigation, collision detection/avoidance for small scale aerial robots, control of multiple coupled helicopters, distributed real-time systems.

■ **German Aerospace Center (DLR) -Germany**

Institute of Flight Systems

Autonomous Rotorcraft Testbed for Intelligent Systems - ARTIS

<http://www.dlr.de/ft/>

Research focused on the development and integration of technologies and

components for autonomous flight. It includes works on controller design, testing of human-Machine interfaces, implementation of intelligent behavior, Sensor fusion, image processing for collision avoidance or optical navigation among others.

■ **Linköping University - Sweden**

Unmanned Aircraft System Technologies Lab UASTech

<http://www.ida.liu.se/divisions/aiics/aiicssite/uastech/>

The UASTech lab has a multi-disciplinary group with backgrounds in control theory, aeronautical engineering, signal processing, computer science, artificial intelligence, and software engineering. Basic and applied research goals include: design and implementation of integrated control modes for UAVs. These include autonomous take-off and landing, trajectory following and vehicle following.

■ **Universidad de Minho -Portugal**

AIVA project .

http://aiva.dei.uminho.pt/aiva_ing/index.htm

The purpose of this project is the construction and automatization of a small airplane, capable to fly autonomously and guided by vision that allows it to operate in any atmospheric condition and recurring (only) to the on-board systems and processing capabilities, to perform various kinds of applications, such as: Monitoring of forests and natural parks (including fire detection), beaches (coastal zones), rivers roads and highways, and of railway lines among others. Support to telecommunications and data transmission and Acquisition of aerial images for territory mapping. design and implementation of sensor platforms for integrated perceptive capability in our UAV platforms, including image processing components and sensor and information fusion techniques

■ **Instituto Superior de Engenharia do Porto - Autonomous systems Laboratory LSA - Portugal.**

<http://www.lsa.isep.ipp.pt>

LSA conducts research in autonomous systems and related areas such as navigation, control and coordination of multiple robots, developed multiple land, air and sea autonomous robots. On the field of UAV their research is focused on systems development for low altitude applications such as forest fire prevention, security, environmental monitoring or aerial imagery.

■ **University of Bristol - Dynamics and Systems Research Group - UK.**

<http://www.bristol.ac.uk/engineering/research/dynamicscontrol/>

The dynamics and control research activity is concerned with research problems relating to modeling, simulation and control of civil, mechanical

and aerospace engineering systems. On the field on Aerial Robotics research involves autonomous landing on moving platforms and aerial refueling among others.

■ **École Polytechnique Fédérale de Lausanne-Laboratory of Intelligent Systems LIS - Switzerland.**

<http://lis.epfl.ch/>

The Laboratory of Intelligent Systems conduct researches on bio-inspired UAVs and visual sensor. They have works on MicroUAV, fast networking using aerial vehicles and bio-inspired vision for direct flying control.

■ **Université de Technologie Campiègne - Heuristic and Diagnosis for Complex Systems Joint research unit HEUDIASYC - France.**

<http://www.hds.utc.fr>

The objective of this research group is the development and control of autonomous aerial vehicles or UAVs focusing on designing, modeling and controlling mini-helicopters such as multi-rotors. Studies are recently devoted to convertible drones capable of vertical take-off and landing as VTOL vehicles and forward flight like planes. The works include technological supports and developments for the design of embedded architectures.

■ **Swiss Federal Institute of Technology, Zurich - Switzerland**

1. **UAV group - Measurement and Control Laboratory.- ETH.**

<http://www.uav.ethz.ch/>

Research activities on flight control for fixed wing aircraft and airships as well as integrated navigation algorithms and computer board developments.

2. **AIRobots, Skysailor, sFly, Flying Reel and my Copter Project - Autonomous System Laboratory.- ETH.**

<http://http://www.asl.ethz.ch/>

Interest is in the development of intelligent products and systems with special emphasis on autonomous mobile robots. AiRobots Goal is to develop a new generation of aerial service robots capable to support human beings in all those activities which require the ability to interact actively and safely with environments not constrained on ground but, indeed, freely in air. The Skysailo goal project is to design and build a solar powered micro airplane for autonomous exploration

3. **UAV Photogrammetry - Institute of Geodesy and Photogrammetry- ETH.**

<http://www.igp.ethz.ch/photogrammetry/>

The main goal of this project is the development and implementation of processing methods of GPS/INS and image data for updating the

position and attitude of the UAV.

■ **Shenyang Institute of Automation, Chinese Academy of Sciences (SIA, CAS)**

<http://www.sia.ac.cn/en/>

■ **Korea Advanced Institute of Science and Technology (KAIST)**

Unmanned System Research Group (FDCL)

<http://unmanned.kaist.ac.kr/>

Interested in the research and development of highly advanced autonomous aerial robots by combining various principles of control theory, aerospace engineering, and computer science

■ **The Australian Research Centre for Aerospace Automation (ARCAA)- Australia**

<http://www.arcaa.aero/>

Conducts research into all aspects of aviation automation, with a particular research focus on autonomous technologies which support the more efficient and safer utilization of airspace, and the development of autonomous aircraft and on-board sensor systems for a wide range of commercial applications.

■ **University of California at Berkley**

1. BEAR: Berkeley Aerobot Team

<http://robotics.eecs.berkeley.edu/bear/>

The BErkeley AeRobot (BEAR) project is a collective, interdisciplinary research effort at UC Berkeley that encompasses the disciplines of hybrid systems theory, navigation, control, computer vision, communication, and multi-agent coordination, since 1996. They currently operate six fully instrumented helicopters, in addition to many fixed- and rotary wing vehicles under development, equipped with GPS/INS, camera, and other sensors on board, which we have been using to validate our control systems design algorithms for UAVs.

2. Center for Collaborative Control of Unmanned Vehicles

<http://c3uv.berkeley.edu/>

interdisciplinary group focused on the fundamental theoretical developments necessary to allow teams of unmanned vehicles to operate autonomously, without extensive monitoring and intervention by human operator

■ **Massachusetts Institute of Technology MIT -USA**

Aerospace Controls Laboratory

<http://acl.mit.edu/>

The Aerospace Controls Laboratory (ACL) researches topics related to autonomous systems and control design for aircraft, spacecraft, and ground

vehicles. Theoretical research is pursued in areas such as: decision making under uncertainty; path planning, activity and task assignment; estimation and navigation; sensor network design; robust control, adaptive control, and model predictive control.. More interesting:

1. UAV SWARM Health Management Project

<http://vertol.mit.edu/prjinfo.html>

Investigating on techniques that will enable the execution of continuous (24-7) mission operations using multiple autonomous vehicles (i.e., vehicle SWARMS) in a dynamic environment.

2. Autonomous UAV Aerobatics Project

<http://aerobatics.mit.edu>

uses a Vicon motion capture sensing to enable rapid prototyping of aerobatic flight controllers for helicopters and aircraft; robust coordination algorithms for multiple helicopters; and vision-based sensing algorithms for indoor flight.

■ **University of Pennsylvania - USA**

The General Robotics, Automation, Sensing and Perception (GRASP) Laboratory

<http://www.grasp.upenn.edu/>

GRASP researchers are building autonomous vehicles and robots, developing self-configuring humanoids, and making robot swarms a reality. Some of the GRASP project are:

1. Micro Autonomous Systems Technologies (MAST)

<http://alliance.seas.upenn.edu/~kumar/wiki/index.php>

Investigating on techniques that will enable the execution of continuous (24-7) mission operations using multiple autonomous vehicles (i.e., vehicle SWARMS) in a dynamic environment.

2. Autonomous Aerial Vehicles

<http://uav-planning.no-ip.org:8080/uav-planning/wiki/WikiStart>

This research project is mainly focused around autonomous navigation of unmanned air vehicles. The challenge is to design systems, which exhibit a goal-driven behavior, while sensing and reacting to changing environment. This project is a collaboration between students and faculty from University of Pennsylvania and industry experts from Dragonfly Pictures, Inc.

■ **Georgia Institute of Technology - USA**

1. Georgia Tech Aerial Robotics Team (GTAR)

<http://controls.ae.gatech.edu/wiki/gtar/>

The Georgia Tech Aerial Robotics Team (GTAR) is a team of students, faculties, and staffs of Georgia Institute of Technology who represents the Institute in the International Aerial Robotics Competition (IARC). The GTAR team has been participating in IARC since its debut in 1991, and also hold the record of winning the most missions. On 2010, the team participated on IARC's 6th mission which is focused on GPS-denied indoor flight. Finished as the leading entry in 2010, the team was able to develop an indoor flight vehicle that can autonomously enter and explore indoor environment. The team plans to complete the mission in the 2011 competition

2. UAV Research Facility (UAVRF)

<http://controls.ae.gatech.edu/wiki/uavrf>

The UAV Research Facility (UAVRF) at Georgia Tech does research related to Unmanned Aerial Vehicles (UAV). The UAVRF operates several different vehicles and conducts flight tests to validate research findings.

■ North Carolina State University

NCSU Aerial Robotics Club

<http://art1.mae.ncsu.edu/>

The NCSU Aerial Robotics Club (ARC) is a student organization that constructs airplanes and puts computers on them to make them intelligent and autonomous. The club aims to participate in the AUVSI and IARC competitions. In common to both competitions is the ability of the airplane to fly autonomously, take high resolution surveillance pictures, and send these pictures back to the ground station.

■ University of Maryland

Autonomous Vehicle Laboratory

<http://www.avl.umd.edu/index.html>

The Autonomous Vehicle Laboratory (AVL) conducts research and development in the area of biologically inspired robotics. They seek to distill the fundamental sensing and feedback principles that govern locomotive behavior in small organisms that will enable the next generation of autonomous microsystems. Capabilities include rapid-prototyping facilities for microsystem fabrication and development, a VICON marker-based visual tracking system that provides direct measurements of 6-DOF vehicle position and orientation for system identification and real-time feedback, a low speed wind tunnel with a specialized high speed camera system for insect tracking and wing kinematics measurement, and advanced hardware and software tools for visual-based simulation of flight systems.

■ University of Southern California

Autonomous Flying Vehicle Project

<http://www-robotics.usc.edu/~avatar/>

The USC Autonomous Flying Vehicle Project was initiated in 1991. Since then the Robotic Embedded Systems Laboratory has designed, built and conducted research with four robot helicopters, the latest being the 3rd generation AVATAR (Autonomous Vehicle Aerial Tracking And Reconnaissance). Since the beginning of the project, a guiding design philosophy has been to create flying robots with high levels of autonomy. Initially, the focus research was in creating a reliable control mechanism for a model helicopter. Once that had been achieved they focused on performing higher lever tasks with the helicopter. Besides stable autonomous flight, they are able to perform tasks such as GPS waypoint navigation, autonomous vision-based landing and autonomous sensor deployment. Researching areas includes autonomous landing on a moving target, deployment on a moving target, stealthy target pursuit and vision-based obstacle avoidance in 3D..

■ **Oakland University-USA**

Embedded System Research Laboratory

<https://sites.google.com/a/oakland.edu/oar/Home>

Research on methodologies, design, verification, and implementation of embedded systems. Current work focuses is on online reconfiguration as means to fault-tolerance for real-time safety-critical distributed embedded systems. Other interests and experiences are in the areas of rapid prototyping, product development, data acquisition and control, Biomedical and autonomous aerial vehicle systems.

■ **Virginia Tech-USA**

Unmanned Systems Lab

<http://www.me.vt.edu/unmanned/index.html>

Research dedicated to autonomous and remotely operated systems development and integration. Areas of expertise include air and ground vehicle design, ground control stations, vision and LIDAR systems, image and signal processing, communications vehicle testing and acoustics among others.

Bibliography

- Achtelik, M., Achtelik, M., Weiss, S., and Siegwart, R. (2011). Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3056–3063.
- Adrien, A., Filliat, D., Doncieux, S., and Meyer, J. (2006). 2d simultaneous localization and mapping for micro air vehicles. In *European Micro Air Vehicle Conference and Flight Competition*.
- Amidi, O. (1996). *An Autonomous Vision-Guided Helicopter*. PhD thesis, Robotics Institute, Carnegie Mellon University.
- Andert, F., Adolf, F.-M., Goormann, L., and Dittrich, J. (2010). Autonomous vision-based helicopter flights through obstacle gates. *Journal of Intelligent and Robotic Systems*, 57:259–280.
- Antonisse, H. J. (1982). Image segmentation in pyramids. *j. Comput Graphics and Image Process*, 19(4):367–383.
- ARCAA, CSIRO-ICT, and QUT-SES (2010). Australian research centre for aerospace automation ARCAA. <http://www.arcaa.aero/>.
- Artieda, J., Sebastian, J. M., Campoy, P., Correa, J. F., Mondragon, I., Martinez, C., and Olivares, M. (2009). Visual 3-d slam from UAVs. *Journal of Intelligent and Robotic Systems*, 55(4):299–321.
- Baker, S. and Matthews, I. (2002). Lucas-kanade 20 years on: A unifying framework: Part 1. Technical Report CMU-RI-TR-02-16, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

- Baker, S. and Nayar, S. K. (1999). A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):1 – 22.
- Barreto, J. a. and Araujo, H. (2001). Issues on the geometry of central catadioptric image formation. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2:II–422–II–427 vol.2.
- Barreto, J. a. and Araujo, H. (2002). Geometric properties of central catadioptric line images. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 237–251, London, UK. Springer-Verlag.
- Beard, R. W. (2007). State estimation for micro air vehicles. In *Innovations in Intelligent Machines (1)*, pages 173–199. Springer Publishing Company, Incorporated.
- Beis, J. S. and Lowe, D. G. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 1000, Washington, DC, USA. IEEE Computer Society.
- Beyeler, A., Zufferey, J.-C., and Floreano, D. (2009). Vision-based control of near-obstacle flight. *Autonomous Robots*, 27(3):201–219.
- Blyenburgh, P. v. (2006). Uav systems : Global review. In *Avionics'06 Conference*, Amsterdam, The Netherlands.
- Bonin-Font, F., Ortiz, A., and Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53:263–296.
- Bosse, M. (1997). A vision augmented navigation system for an autonomous helicopter. Master's thesis, Boston University.
- Bouguet Jean Yves (1999). Pyramidal implementation of the lucas-kanade feature tracker. Technical report, Intel Corporation. Microprocessor Research Labs, Santa Clara, CA 95052.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA.
- Bradski, G. R. (1998). Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 1(Q2).
- Buenaposada, J. M., Munoz, E., and Baumela, L. (2003). Tracking a planar patch by additive image registration. In *Proc. of International workshop, VLBV 2003*, volume 2849 of *LNCS*, pages 50–57.

- Caballero, F., Merino, L., Ferruz, J., and Ollero, A. (2009). Vision-based odometry and slam for medium and high altitude flying uavs. *Journal of Intelligent and Robotic Systems*, 54:137–161.
- Campoy, P., Correa, J. F., Mondragón, I., Martínez, C., Olivares, M., Mejías, L., and Artieda, J. (2009). Computer vision onboard UAVs for civilian tasks. *Journal of Intelligent and Robotic Systems*, 54(1-3):105–135.
- Capel, D. P. (2001). *Image Mosaicing and Super-resolution*. PhD thesis, University of Oxford, Robotics Research Group, Department of Engineering Science.
- Carnie, R., Walker, R., and Corke, P. (2006). Image processing algorithms for UAV "sense and avoid". In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2848–2853.
- Cesetti, A., Frontoni, E., Mancini, A., Ascani, A., Zingaretti, P., and Longhi, S. (2011). A visual global positioning system for unmanned aerial vehicles used in photogrammetric applications. *Journal of Intelligent and Robotic Systems*, 61:157–168.
- Cesetti, A., Frontoni, E., Mancini, A., Zingaretti, P., and Longhi, S. (2010). A vision-based guidance system for uav navigation and safe landing using natural landmarks. *Journal of Intelligent and Robotic Systems*, 57:233–257.
- Chaumette, F. and Hutchinson, S. (2006). Visual servo control. I. basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90.
- Chaumette, F. and Hutchinson, S. (2007). Visual servo control, part II: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118.
- Cheng, Y., Maimone, M. W., and Matthies, L. (2006). Visual odometry on the mars exploration rovers. *IEEE Robotics and Automation magazine*, 13(2):54–62.
- Conroy, J., Gremillion, G., Ranganathan, B., and Humbert, J. S. (2009). Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. *Autonomous Robots*, 27(3):189–198.
- Corke, P. and Hutchinson, S. (2001). A new partitioned approach to image-based visual servo control. *Robotics and Automation, IEEE Transactions on*, 17(4):507 –515.
- Corke, P., Sikka, P., and Roberts, J. (2000). Height estimation for an autonomous helicopter. In *Proc. Int. Symp. Experimental Robotics*, Hawaii.
- Corke, P., Strelow, D., and Singh, S. (2004). Omnidirectional visual odometry for a planetary rover. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Japan.

- Corke, P. I. (2010). Spherical image-based visual servo and structure estimation. In *Proceedings 2010 IEEE International Conference on Robotics and Automation (ICRA)*.
- Cornall, T. and Egan, G. (2004). Measuring horizon angle from video on a small unmanned air vehicle. In *2nd International Conference on Autonomous Robots and Agents*, New Zealand.
- Cornall, T., Egan, G., and Price, A. (2006). Aircraft attitude estimation from horizon video. *Electronics Letters*, 42(13):744–745.
- Criminisi, A., Reid, I. D., and Zisserman, A. (1999). A plane measuring device. *Image Vision Comput.*, 17(8):625–634.
- Cui, H., Lin, Z., and Zhang, J. (2008). Research on low altitude image acquisition system. In Li, D., editor, *Computer And Computing Technologies In Agriculture, Volume I*, IFIP International Federation for Information Processing, pages 95–102. Springer Boston.
- CVG-UPM (2010). Universidad Politécnica de Madrid. Computer Vision Group. Vision for UAV Project. <http://www.vision4uav.com>.
- Dalamagkidis, K., Valavanis, K., and Piegl, L. (2008). Current status and future perspectives for unmanned aircraft system operations in the us. *Journal of Intelligent and Robotic Systems*, 52:313–329. 10.1007/s10846-008-9213-x.
- Davison, A., Reid, I., Molton, N., and Stasse, O. (2007). Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052 –1067.
- De Wagter, C. and Mulder, J. (2005). Towards vision-based uav situation awareness. *AIAA Guidance, Navigation, and Control Conference and Exhibit*.
- Demonceaux, C., Vasseur, P., and Pđžgard, C. (2006). Omnidirectional vision on UAV for attitude computation. In *IEEE International Conference on Robotics and Automation 2006 (ICRA'06)*, pages 2842–2847, Orlando, FL, US. IEEE.
- Dobrokhodov, V., Kaminer, I., Jones, K., and Ghabcheloo, R. (2006). Vision-based tracking and motion estimation for moving targets using small uavs. *American Control Conference, 2006*, pages 6 pp.–.
- Dumai, A. and Winkler, A. (1990). Reliability prediction model for gyroscopes. *Annual Reliability and Maintainability Symposium, 1990. Proceedings.*, pages 5–9.
- Dunbabin, M., editor (2007). *A Vision System for Optic-flow-based Guidance of UAVs*, Sydney, Australia. Australian Robotics and Automation Association (ARAA).

- Dusha, D., Boles, W., and Walker, R. (2007). Fixed-wing attitude estimation using computer vision based horizon detection. In *In Proceedings 12th Australian International Aerospace Congress*, pages 1–19, Melbourne, Australia.
- Egbert, J. and Beard, R. W. (2010). Low-altitude road following using strap-down cameras on miniature air vehicles. *Mechatronics*, In Press, Corrected Proof:–.
- Eisenbeiss, H. (2004). A mini unmanned aerial vehicle (uav): system overview and image acquisition. In *Image Acquisiton” International Workshop on Processing and Visualization using High-Resolution Imagery*.
- Eisenbeiss, H. (2009). *UAV Photogrammetry*. PhD thesis, Institut für Geodäsie und Photogrammetrie Eidgenössische Technische Hochschule Zürich, Wolfgang-Pauli-Strasse 15 8093 Zürich.
- Eng, P., Mejias, L., Liu, X., and Walker, R. (2010). Automating human thought processes for a uav forced landing. *Journal of Intelligent and Robotic Systems*, 57:329–349.
- Ettinger, S. M., Nechyba, M. C., Ifju, P. G., and Waszak, M. (2002). Vision-guided flight stability and control for micro air vehicles. In *IEEE International Conference on Intelligent Robots and Systems*. IEEE.
- European Commission, D. (2009). Hearing on light unmanned aircraft systems (UAS). Technical report, European Commission , Directorate General Mobility and Transport.
- European Commission, E. (2007). Study analysing the current activities in the field of UAV, entr/2007/065. Technical report, European Commission, Enterprise industry Directorate General.
- Fischer, M. A. and Bolles, R. C. (1981). Random sample concensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- Frost and Sullivan (2008). European civil and commercial UAV markets. Technical report, Frost & Sullivan.
- Fucen, Z., Haiqing, S., and Hong, W. (2009). The object recognition and adaptive threshold selection in the vision system for landing an unmanned aerial vehicle. In *Information and Automation, 2009. ICIA '09. International Conference on*, pages 117 –122.
- Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32 – 40.

- Geyer, C. and Daniilidis, K. (2000). A unifying theory for central panoramic systems and practical applications. In *ECCV* (2), pages 445–461.
- Geyer, C. and Daniilidis, K. (2001). Catadioptric projective geometry. *Journal of Computer Vision*, 43:223–243.
- Golightly, I. and Jones, D. (2005). Visual control of an unmanned aerial vehicle for power line inspection. In *Advanced Robotics, 2005. ICAR '05. Proceedings., 12th International Conference on*, pages 288–295.
- Haddon, D. R. and Whittaker, C. J. (2004). Uk-caa policy for light uav systems. Technical report, UK Civil Aviation UK Civil Aviation Authority, London, UK.
- Harris, C. G. and Stephens, M. (1988). A combined corner and edge detection. In *In Proceedings of the 4th Alvey Vision Conference*, pages 147–151.
- Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.
- He, Z., V, L. R., and R, C. P. (2006). Vision-based UAV flight control and obstacle avoidance. In *Proceedings of the American Control Conference*, page 5pp.
- Herbert Bay, Tinne Tuytelaars, and Luc Van Gool (2006). SURF: Speeded Up Robust Features. In *Proceedings of the ninth European Conference on Computer Vision*.
- Hrabar, S. and Sukhatme, G. (2003). Omnidirectional vision for an autonomous helicopter. In *In IEEE Internation Conference on Robotics and Automation*, pages 558–563.
- Hrabar, S., Sukhatme, G., Corke, P., Usher, K., and Roberts, J. (2005). Combined optic-flow and stereo-based navigation of urban canyons for a uav. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3309–3316.
- Hutchinson, S., Hager, G. D., and Corke, P. (1996). A tutorial on visual servo control. In *IEEE Transaction on Robotics and Automation*, volume 12(5), pages 651–670.
- ICPUAS, UPM-CVG, ARCAA, and Cranfield University (2009). International cooperation program for unmanned aerial systems research and development ICPUAS. <http://www.vision4uav.com>.
- Irani, M. and Anandan, P. (2000). About direct methods. In Triggs, B., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 267–277. Springer Berlin HeidelbergHeidelberg.

- JAA Eurocontrol, J. (2004). A concept for european regulations for civil unmanned aerial vehicles (UAV). final report. Technical report, Joint JAA Eurocontrol Initiative on UAVs.
- Johnson, A. and Matthies, L. (1999). Precise image-based motion estimation for autonomous small body exploration. In *5th International Symp. on Artificial Intelligence, Robotics and Automation in Space*, pages 627–634.
- Kendoul, F., Nonami, K., Fantoni, I., and Lozano, R. (2009). An adaptive vision-based autopilot for mini flying machines guidance, navigation and control. *Autonomous Robots*, 27(3):165–188.
- Labrosse, F. (2006). The visual compass: Performance and limitations of an appearance-based method. *Journal Of Field Robotics*, 23(10):913–941.
- Lai, J., Mejias, L., and Ford, J. J. (2011). Airborne vision-based collision-detection system. *Journal of Field Robotics*, 28(2):137–157.
- Lange, S., Sünderhauf, N., and Protzel, P. (2008). Autonomous landing for a multirotor UAV using vision. In *In Workshop Proceedings of SIMPAR Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS*, pages 482–491, Venice, Italy.
- Lewis, G. (2007). Evaluating the use of a low-cost unmanned aerial vehicle platform in acquiring digital imagery for emergency response. In Li, J., Zlatanova, S., and Fabbri, A. G., editors, *Geomatics Solutions for Disaster Management*, Lecture Notes in Geoinformation and Cartography, pages 117–133. Springer Berlin Heidelberg.
- Lin, F., Lum, K.-Y., Chen, B., and Lee, T. (2009). Development of a vision-based ground target detection and tracking system for a small unmanned helicopter. *Science in China Series F: Information Sciences*, 52:2201–2215.
- Liu, Y.-c. and Dai, Q.-h. (2010). A survey of computer vision applied in aerial robotic vehicles. In *Optics Photonics and Energy Engineering (OPEE), 2010 International Conference on*, volume 1, pages 277 –280.
- Lowe, D. G. (2004). Distintive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proc. of the 7th IJCAI*, pages 674–679, Vancouver, Canada.
- Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 D Visual Servoing. *IEEE Trans. on Robotics and Automation*, 15:238–250.
- Matthies, L. (1989). *Dynamic stereo vision*. Cmu-cs-89-195, Carnegie Mellon University. Computer Science Department.

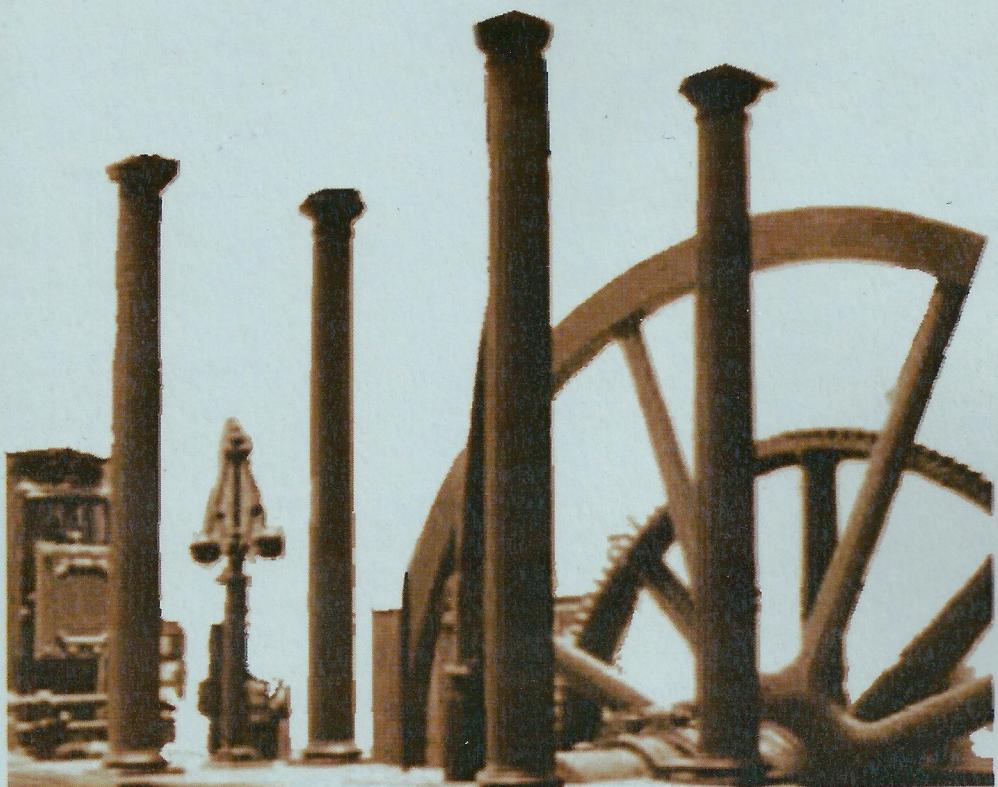
- Mei, C. and Rives, P. (2007). Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*.
- Mejias, L., Campoy, P., Mondragon, I., and Doherty, P. (2007). Stereo visual system for autonomous air vehicle navigation. In *6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV 07)*, pages 0–0, Toulouse, France.
- Mejias, L., Saripalli, S., Campoy, P., and Sukhatme, G. (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., Martínez-de Dios, J., Ferruz, J., and Ollero, A. (2006). A cooperative perception system for multiple uavs: Application to automatic detection of forest fires. *J. Field Robotics*, 23(3-4):165–184.
- Merz, T., Duranti, S., and Conte, G. (2004). Autonomous landing of an unmanned helicopter based on vision and inertial sensing. In *International Symposium on Experimental Robotics*, Singapore.
- Merz, T., Duranti, S., and Conte, G. (2006). Autonomous landing of an unmanned helicopter based on vision and inertial sensing. In Ang, M. and Khatib, O., editors, *Experimental Robotics IX*, volume 21 of *Springer Tracts in Advanced Robotics*, pages 343–352. Springer Berlin / Heidelberg.
- Metni, N. and Hamel, T. (2007). A uav for bridge inspection: Visual servoing control law with orientation limits. *Automation in Construction*, 17(1):3–10.
- Mikolajczyk, M. and Smid, C. (2005). A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.
- Montiel, J., Civera, J., and Davison, A. (2006). Unified inverse depth parametrization for monocular slam. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA.
- Myers, J. S. and Miller, R. L. (2005). Optical airborne remote sensing. In Miller, R. L., Del Castillo, C. E., and McKee, B. A., editors, *Remote Sensing of Coastal Aquatic Environments*, volume 7 of *Remote Sensing and Digital Image Processing*, pages 51–67. Springer Netherlands.
- Nayar, S. and Baker, S. (1997). A theory of catadioptric image formation. Technical Report CUCS-015-97, Department of Computer Science, Columbia University.
- Nemra, A. and Aouf, N. (2009). Robust airborne 3d visual simultaneous localization and mapping with observability and consistency analysis. *Journal of Intelligent and Robotic Systems*, 55:345–376.

- Nikolos, I. K., Tsourveloudis, N. C., and Valavanis, K. P. (2004). A uav vision system for airborne surveillance. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, pages 77–83, New Orleans, LA, USA.
- Nistér, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal Of Field Robotics*, 23(1):3–20.
- Nonami, K., Kendoul, F., Suzuki, S., Wang, W., Nakazawa, D., Nonami, K., Kendoul, F., Suzuki, S., Wang, W., and Nakazawa, D. (2010). Guidance and navigation systems for small aerial robots. In *Autonomous Flying Robots*, pages 219–250. Springer Japan.
- Okamoto Jr, J. and Grassi Jr., V. (2002). Visual servo control of a mobile robot using omnidirectional vision. In *Proceedings of Mechatronics 2002*, pages 413–422, University of Twente, Netherlands.
- Olivares-Méndez, M., , Mondragón, I. F., Campoy, P., and Martínez, C. (2010). Fuzzy controller for uav-landing task using 3d-position visual estimation. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1 –8.
- Ollero, A. and Merino, L. (2004). Control and perception techniques for aerial robotics. *Annual Reviews in Control*, 28(2):167 – 178.
- Otsu, N. (1979). A threshold selection method from gray level histograms. In *IEEE Trans. Syst. Man Cybern.*, volume 9, pages 62–66.
- Puri, A., Valavanis, K., and Kontitsis, M. (2007). Statistical profile generation for traffic monitoring using real-time uav based video data. In *Control and Automation, 2007.MED '07. Mediterranean Conference on*, MED, pages 1–6.
- Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M., and Rinner, B. (2010). Networked uavs as aerial sensor network for disaster management applications. *i Elektrotechnik und Informationstechnik*, 127:56–63.
- Rathinam, S., Kim, Z., and Sengupta, R. (2006). Vision-based following of structures using an unmanned aerial vehicle. Technical Report UCB-ITS-RR-2006-1, University of California, Berkeley.
- Roumeliotis, S. I., Johnson, A., and Montgomery, J. F. (2002). Augmentation inertial navigation with image-based motion estimation. In *IEEE International Conference on Robotics and Automation*, pages 4326–4333, Washington D.C.
- Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA.

- Salazar, S., Romero, H., Gomez, J., and Lozano, R. (2009). Real-time stereo visual servoing control of an uav having eight-rotors. In *Electrical Engineering, Computing Science and Automatic Control, CCE, 2009 6th International Conference on*.
- Sanderson, A. C. and Weiss, L. E. (1983). Adaptative visual servo control of robots. In *Robot Vision* (A. Pugh, ed), pages 107–116.
- Saripalli, S., Montgomery, J. F., and Sukhatme, G. S. (2003). Visually-guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–381.
- Saripalli, S. and Sukhatme, G. S. (2007). Landing a helicopter on a moving target. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2030–2035, Rome, Italy.
- Scaramuzza, D., Martinelli, A., and Siegwart, R. (2006). A toolbox for easy calibrating omnidirectional cameras. In *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*.
- Scaramuzza, D. and Siegwart, R. (2008). Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics, Special Issue on Visual SLAM*. In press. Guest editors: Jose’ Neira, Andrew Davison, John J. Leonard, Publication date: October 2008.
- Shi, J. and Tomasi, C. (1994). Good features to track. In *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’94)*, pages 593–600.
- Siciliano, B. and Khatib, O., editors (2008). *Springer Handbook of Robotics*. Springer, Berlin, Heidelberg.
- Simon, G. and Berger, M.-O. (2002). Pose estimation for planar structures. *Computer Graphics and Applications, IEEE*, 22(6):46–53.
- Simon, G., Fitzgibbon, A., and Zisserman, A. (2000). Markerless tracking using planar structures in the scene. In *Augmented Reality, 2000. (ISAR 2000). Proceedings. IEEE and ACM International Symposium on*, pages 120–128.
- Sinha, S. N., michael Frahm, J., Pollefeys, M., and Genc, Y. (2006). Gpu-based video feature tracking and matching. In *In Workshop on Edge Computing Using New Commodity Architectures*.
- Sturm, P. (2000). Algorithms for plane-based pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, pages 1010–1017.
- Svoboda, T. and Pajdla, T. (2002). Epipolar geometry for central catadioptric cameras. *Int. J. Computer Vision.*, 49(1):23–37.

- Svoboda, T., Pajdla, T., and Hlaváč, V. (1997). Central panoramic cameras: Geometry and design. Research report K335/97/147, Czech Technical University, Faculty of Electrical Engineering, Center for Machine Perception, FEL ČVUT, Karlovo náměstí 13, Praha, Czech Republic. Available at <ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/TR-K335-97-147.ps.gz>.
- ThomsonReuters (2010). 2010 Journal Citation Reports JCR science edition. Technical report, Thomson Reuters.
- Thurrowgood, S., Soccol, D., Moore, R., Bland, D., and Srinivasan, M. (2009). A vision based system for attitude estimation of uavs. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5725 –5730.
- Todorovic, S., Nechyba, M., and Ifju, P. (2003). Sky/ground modeling for autonomous MAV flight. *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, 1:1422–1427.
- Valavanis, K. P. and Kontitsis, M. (2007). A historical perspective on unmanned aerial vehicles. In Valavanis, K. P., editor, *Advances in Unmanned Aerial Vehicles*, volume 33 of *Intelligent Systems, Control and Automation: Science and Engineering*, pages 15–46. Springer Netherlands.
- Vicon Motion Systems (2011). VICON MX digital optical motion capture system. <http://www.vicon.com>.
- Walraven, J. A. (2003). Failure mechanisms in M.E.M.S. *Internantional Test Conference*, page 828.
- Weibel, R. E. and Hansman, R. J. (2009). Assuring safety through operational approval: Challenges in assessing and approving the safety of systems-level changes in air transportation, ICAT:2009-04. Technical report, MIT International Center for Air Transportation (ICAT), Massachusetts Institute of Technology,Cambridge, MA 02139 USA.
- Weiss, L. (1984). *Dynamic Visual Servo Control of Robots: An Adaptive Image-Based Approach*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- Welch, G. and Bishop, G. (1995). An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.
- Welch, G. and Bishop, G. (2005). European civil unmanned air vehicle roadmap, volume 3 strategic research agenda. Technical report, European Civil UAV FP5 R&D Program Members.
- Wenzel, K., Masselli, A., and Zell, A. (2011). Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle. *Journal of Intelligent and Robotic Systems*, 61:221–238.

- Ying, X. and Hu, Z. (2004a). Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 442–455. Springer Berlin / Heidelberg.
- Ying, X. and Hu, Z. (2004b). Catadioptric camera calibration using geometric invariants. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(10):1260–1271.
- Zach, C., Gallup, D., and Frahm, J.-M. (2008). Fast gain-adaptive klt tracking on the gpu. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1 –7.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334.
- Zou, H., Gong, Z., Xie, S., and Ding, W. (2006). A pan-tilt camera control system of uav visual tracking based on biomimetic eye. *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, pages 1477–1482.



**ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS INDUSTRIALES**

UNIVERSIDAD POLITÉCNICA DE MADRID