
POLITECNICO DI TORINO

**Master of Science in Mechatronic Engineering
(Ingegneria Meccatronica)**



Master's Thesis

POSE ESTIMATION FOR UNMANNED AERIAL VEHICLES BASED ON CONIC CORRESPONDENCES

Supervisors:

Prof. Carlo Novara

Prof. Erjen Lefeber

Student:

Ivan Domenico Barbieri

Academic Year

2019/2020

Oh Madre Vita.

*Unico respiro che mi accompagna dal primo dei giorni,
unica torcia che illumina l'imprevedibile sentiero,
unico corroborante che foraggia i miei desideri più profondi,
e unica mano, che guida il mio passo primiero.*
*Tu, Madre, non lasciar che i miei occhi osservino solo questi mondi,
ma esorta il pigro sguardo a squarciare il velo del vero,
per poter sbirciare, attraverso i gloriosi tagli,
il Mondo occupato da miei Dei e Angeli.*

Dedicato a chi soffre...

This thesis wouldn't have
been possible without
the collaboration with
Technische Universiteit Eindhoven (TU/e).
I would particularly like to thank
Dr. Erjen Lefeber for giving me this opportunity
and for his time spent with me.



TU/e

Contents

List of Figures VII

1	Introduction	1
1.1	Abstract of the Thesis	4
1.2	Autonomous Vehicle	2
1.3	Unmanned Aerial Vehicle	3
1.4	Outline	5

I Pose Estimation using Homography

2	Homography Estimation Using Point Correspondences	9
2.1	Theoretical Homography Matrix Computation	10
2.2	Experimental Homography Matrix Estimation Using Points’ Correspondences	12
2.3	Tests’ Results	14
3	Homography Estimation Using Conic Correspondences	15
3.1	Conics: Basic Concepts	16
3.1.1	Conics	16
3.1.2	Ellipsis	17
3.2	Theoretical Homography Matrix Estimation Using Ellipsis’ Correspondences	18

3.3	Experimental Homography Matrix Estimation Using Ellipsis' Correspondences	20
3.4	Tests' Results	23
4	Observer Design for Homography Matrix Estimation	25
4.1	Introduction to State Observer	26
4.2	Nonlinear Observer Design on the Special Linear group $SL(3)$	27
4.3	Tests' Results	32
5	Video Rectification using Homography Estimation	43
5.1	Video Implementation of Static Estimation, , .	44
5.2	Video Implementation of Observer Estimation	47
5.3	Comparison of the two Methods	50
II	Modelling of Parrot-Mambo Drone	
6	Quadrotor Modelling	53
6.1	Quadrotor Mathematical Model	54
6.1.1	Nomenclature	55
6.1.2	State Equations	55
6.1.3	Output Equations	56
6.1.4	Total System	59
6.2	Observability Analysis	61
7	Conclusions	63
7.1	Conclusive Considerations	63
7.2	Recommendations	65

Appendices

A	Proofs and Methods	67
A.1	Single Value Decomposition (SVD)	67
A.2	“Harris Corner Detection” Algorithm	68
A.3	Least Square Method	69
B	Glossary	71
C	Code Lines	73
	References	77
	Epilogue	79

List of Figures

Figure 1.1: Autonomous robot technologies for space operation designed by the DFKI Robotics Innovation Centre develops	3
Figure 1.2: Main Building Blocks of an Autonomous Vehicle	3
Figure 1.3: Northrop Grumman X-47B UAV over the U.S. Navy's Atlantic Test Range	4
Figure 1.4: Mambo Parrot fly mini-drone	4
Figure 2.1: The same point present on a planar surface appear different in two camera plane when they have different position and orientation. The homography matrix H is the link between these two views	10
Figure 2.2: Reference Image used as the first sight in order to estimate the homography matrix	12
Figure 2.3: Reference image after being processed. The blue spots are the detected corners of the first rectangle, the red ones are the detected corners of the second rectangle	13
Test 2.1	14
Figure 3.1: The different types of conic as an intersection of a plane with a double cone	16
Figure 3.2: Ellipse-to-ellipse correspondence defines a local affine transformation A up to an unknown rotation R	18
Figure 3.3: Reference Image used for the homography estimation	20
Figure 3.4: Reference and Moved Image after the Image Processing step	21
Test 3.1	23
Test 3.2	23
Test 3.3	24
Figure 4.1: Block Diagram of the Simulink implementation of the observer's continuous version	30
Test 4.1	33
Test 4.2	34
Test 4.3	35
Test 4.4	36
Test 4.5	37
Test 4.6	38

Test 4.7	39
Test 4.8	40
Figure 5.1: Frame 1 rectification done after using Static Estimator	44
Figure 5.2: Frame 161 rectification done after using Static Estimator	45
Figure 5.3: Frame 244 rectification done after using Static Estimator	46
Figure 5.4: Frame 380 rectification done after using Static Estimator	46
Figure 5.5: Frame 628 rectification done after using Static Estimator	47
Figure 5.6: Frame 1 rectification done after using Static Estimator	48
Figure 5.7: Frame 161 rectification done after using Observer Estimator	48
Figure 5.8: Frame 244 rectification done after using Observer Estimator	48
Figure 5.9: Frame 380 rectification done after using Observer Estimator	49
Figure 5.10: Frame 628 rectification done after using Observer Estimator	49
Figure 6.1: Inertial and mobile reference frames	54
Figure 6.1: Inertial, mobile and camera reference frames	56
Figure 6.2: Model of camera image reconstruction	56
Figure 6.4: Bird view of world space and camera space	57

Chapter 1

Introduction

This is the anchor chapter of the thesis. It has the purpose of introducing and explaining not only the researches done in my work, but also the way the latter are carried on.

The first section contains the abstract of the thesis, where is described the main goal of this work, with a brief overview on the methodologies used.

In the second section, after an initial introduction where is explained the increasing importance and usage of autonomous vehicles in our society, is explained briefly how these vehicle works and the link between all their features.

In the second section, the attention is focused on the principal autonomous vehicles for this research, the UAV (Unmanned Autonomous Vehicle). This paragraph discusses what UAVs are, discussing their main uses in society. Then is explained what a quadrotor is with a sight on the Parrot Mambo fly drone.

Finally, in the fourth one an outline of the following chapters is made.

1.1 Abstract of the Thesis

A crucial point for the automation of a vehicle is the positioning process. There are two types of positioning, the Global positioning (geographic coordination, achieved using a dedicated network of satellites in orbit), and Local positioning (estimating the pose of the vehicle with respect to a local coordinate frame). In this work, the first steps to achieve an online local positioning using a UAV are described.

In particular, the sUAV considered is the Parrot Mambo fly drone. This economic and small drone has limited on board sources, and the choice of sensor to be used cannot be done in a wide range. A peculiarity of this sUAV is that is designed with an on-board camera. This camera is the chosen sensor used in order to achieve the final goal. There are several techniques that could be used for this purpose; the one selected and more suitable is the homography technique. The latter is a technique widely used in computer vision, and takes the name from the homonymous matrix that expresses the geometrical transformation between two images of the same view, with different points of view. In this work the homography technique is presented and developed in different shape, using not only different reference images (rectangular and elliptical shapes), but different technique too (static computation and state observer implementation).

Moreover, in order to make this research applicable to our specific case, a mathematical model of the drone has been developed, with two main peculiarities. The first is the use of the rotation matrix R belonging to the 3D rotation group $SO(3)$ in order to represent the attitude of the quadrotor. The choice of R is preferable than using Euler angles (which have singularities in representation) or quaternions (which might lead to ambiguous control actions due to the phenomenon of unwinding) mainly used in other mathematical quadrotors' models. The second is the choice of the model's output, which is the transformation of a specific point from a fixed reference frame, to the raster coordinate system.

All my researches are introduced with clearly and purely mathematical theory, accompanied by sets of tests and simulations which algorithm has been articulate step by step, and which results are commented and refined with personal comments. The simulation have been done using MATLAB software, which includes the homonymous language.

1.2 Autonomous Vehicle

Autonomous vehicles (or self-driving vehicles) are becoming more and more important in these years. Such a new and emerging technology has the potential of completely reinventing the way we live.

Self-driving vehicles' utilization ranges over all the life aspects, from automation of production in a company, to travelling reason, passing through house utility. For this reason, the scenario in which an autonomous vehicle works could be very different. For example a utility vehicle (e.g. robot vacuum cleaner) moves in a confined and safe location, while a self-driving car faces a more challenging scenario, where interfaces with the environment (people, obstacles, other vehicles) and the use of a predefined path is not possible.

What almost all the autonomous vehicles have in common is their way of “thinking”, which could be summarized in the following building blocks:

- Sensors;
- V2X (Vehicle-to-X communication);
- Actuators;
- Perception;
- Planning;
- Control;

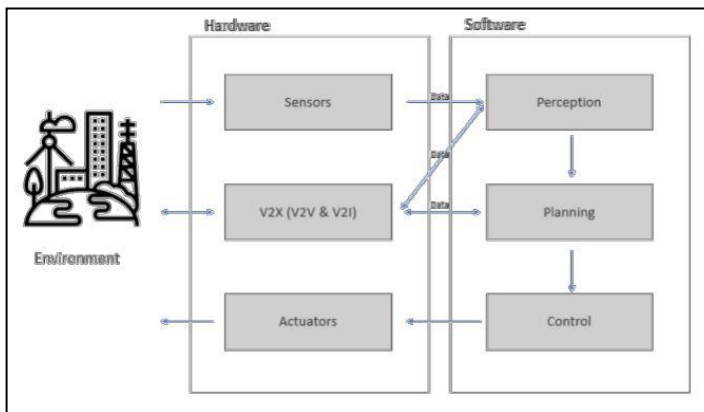


Figure 1.2 – Main Building Blocks of an Autonomous Vehicle.



Figure 1.1 - Autonomous robot technologies for space operation designed by the DFKI Robotics Innovation Centre develops.

The first three blocks of this make up the group of hardware, while the next three ones the group of software.

The sensors (Camera, LiDar, Global Positioning System (GPS), etc.) are the components that allow the autonomous vehicle to take in raw information about the environment. The V2X communication (Vehicle-to-Vehicle or Vehicle-to-Infrastructure) enable the autonomous vehicle to talk to and receive information from other machines or infrastructure agents in the environment. The actuators

are the components of a vehicle responsible for controlling and moving the system.

The perception system refers to the ability of the autonomous vehicle to understand what the raw information coming in through the sensors or V2V components mean. The planning system refers to the ability of the autonomous vehicle to make certain decisions to achieve some higher order goals. This is how the autonomous vehicle knows what to do in a specific situation. The planning system works by combining the processed information about the environment with established policies and knowledge about how to navigate in the environment, so that the system can determine what action to take. Finally, the control system pertains to the process of converting the intentions and goals derived from the planning system into actions. Here the control system sends to the hardware (the actuators) the necessary inputs that will lead to the desired motions.

1.3 Unmanned Aerial Vehicle

An UAV (Unmanned Aerial Vehicle, or Uncrewed Aerial Vehicle), commonly known as a drone, is an aircraft without a human pilot on board and a type of unmanned vehicle. UAVs are a component of an UAS (unmanned aircraft system), which include a UAV, a ground-based controller, and a system of communication between the two. The flight of UAVs may operate with various degrees of autonomy: either under remote control by a human operator or autonomously by on-board computers.



Figure 1.3 - Northrop Grumman X-47B UAV over the U.S. Navy's Atlantic Test Range.

Compared to crewed aircraft, UAVs were originally used for missions too "dull, dirty or dangerous" for humans. While they originated mostly in military applications, their use is rapidly expanding to commercial, scientific, recreational, agricultural, and other applications, such as policing and surveillance, product deliveries, aerial photography, infrastructure inspections, smuggling, and drone racing. Civilian UAVs now vastly outnumber military UAVs, with estimates of over a million sold by 2015.

Recent advances in sensor, in microcomputer technology, in control and in aerodynamics theory have made sUAV (small Unmanned Aerial Vehicle) a reality. The small size, low cost and manoeuvrability of these systems have made them potential solutions in a large class of applications. However, the small size of these vehicles poses significant challenges. The small sensors used on these systems are much noisier than their larger counterparts are. The compact structure of these vehicles also makes them more vulnerable to environmental effects. Since the number and complexity of applications for such systems grows daily, the control techniques involved must also improve in order to provide better performance and increased versatility.



Figure 1.4 - Mambo Parrot fly mini-drone.

My research has the final goal of providing a good positioning system for the Parrot Mambo fly drone. This is a quadcopter drone, a drone made up of four engines, holds the electronic board in the middle and the engines at four extremities. The attitude and position of the quadrotor can be controlled to desired values by changing the speeds of the four motors. The following forces and moments can be performed on the

quadrotor: the thrust caused by rotors rotation, the pitching moment and rolling moment caused by the difference of four rotors thrust, the gravity, the gyroscopic effect, and the yawing moment. The gyroscopic effect only appears in the lightweight construction quadrotor. The yawing moment is caused by the unbalanced of the four rotors rotational speeds. The yawing moment can be cancelled out when two rotors rotate in the opposite direction. So, the propellers are divided in two groups. In each group there are two diametrically opposite motors that we can easily observe thanks to their direction of rotation.

1.4 Outline

This thesis consist of two parts, divided in seven chapters. Each chapter contains the description of the steps followed during the work period of the thesis, and provides its first page a brief recap of the work done in the previous ones with a description of the content.

Part I: Pose Estimation using Homography

The first part of the thesis consist in finding a way to estimate the pose of a Unmanned Aerial Vehicle using the technique of Homography. It is composed by the following chapters.

Chapter 2. In this chapter a qualitative introduction on homography is discussed. Then its computation using point correspondences is discussed, with the presentation of the main tests' results.

Chapter 3. In this chapter an introduction on conic shapes and their way of representation is discussed. Then the computation using conic correspondences is discussed, with the presentation of the main tests' results.

Chapter 4. In this chapter a NonLinear Observer on $SO(3)$ for the homography estimation using point correspondence is developed. Then, the presentation of the main tests' results.

Chapter 5. In this chapter the Static Homography Estimator presented in Chapter 3 and the Observer presented in Chapter 4 are used in order to process an image rectification of a video.

Part II: Modelling of Parrot-Mambo Drone

The second part consist in modelling the Parrot-Mambo drone obtaining a mathematical description of the transformation of a point from a World reference system to the camera picture one.

Chapter 6. In this chapter a qualitative introduction on the principles of working of a quadrotor is discussed. Then the mathematical model is presented and its observability is studied.

Chapter 7. Finally, in this chapter, conclusions and possible future developments of the work are presented.

Finally, the **Appendix A** contains the proof of some theorem and the explanation of some method used in this thesis and **Appendix B** contains a Glossary of the main used variables.

Part I

Pose Estimation using Homography

Chapter 2

Homography Estimation Using Point Correspondences

In the previous chapter was introduced the world of UAV and how its impact on the society is exponentially increasing. Moreover is explained the final goal of this thesis project including a brief explanation of the steps followed. The first part of my research work is focused on the estimation of the homography matrix that links two views of the same scene using point correspondences. This is the easiest way of computing the homography matrix. This estimation makes the transformation possible of an image in order to make it overlap with a reference one.

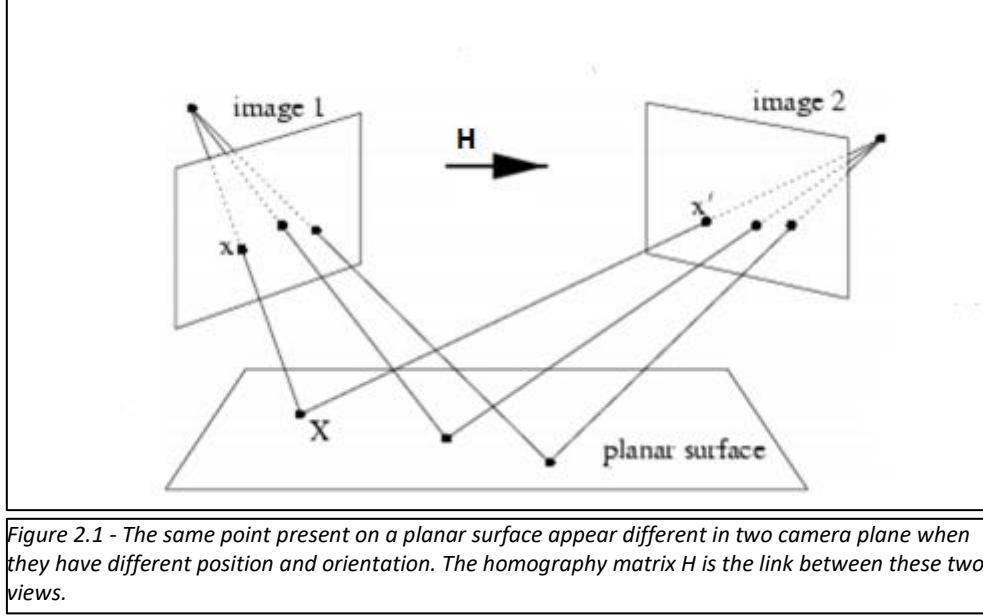
The first section is an introduction to the theory behind the estimation, making clear the analytical process used in the experimental tests.

The second section is describes systematically the algorithm designed by me and used for the estimation of the homography matrix between two views of the same scene making possible the warping of the moved image.

Finally the third section shows and comment on the results from the tests done.

2.1 Theoretical Homography Matrix Computation

The homography could be a projection between 3D-to-2D or between 2D-to-2D. The second case is the one in which we are interested, in particular when it is referred to a projection from one camera to another.



The position of a point in the Camera Coordinate System is obtained by the product:

$$\begin{pmatrix} x_1 \\ y_1 \\ w_1 \end{pmatrix} = [M_{1,in}][M_{1,ex}] \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} = M_1 \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix},$$

where M_{ex} is the extrinsic parameters matrix that expresses the pose of the WCS into the CCS, M_{in} is the intrinsic parameter matrix that transforms the coordinates from the image plane of the camera into the raster plane, $(x_1, y_1, w_1)^T$ are the homogeneous coordinate of the point expressed in the CCS, $(X, Y, Z, W)^T$ are the coordinate of the point expressed in the WCS and M is the transformation matrix.

The matrix M_I belongs to $\mathbb{R}^{3 \times 4}$, but considering a planar scene ($Z = 0$) we can delete the third column making M_I part of $\mathbb{R}^{3 \times 3}$. Moreover, M_I is invertible, so it is possible to write:

$$\begin{pmatrix} X \\ Y \\ W \end{pmatrix} = [M_1]^{-1} \begin{pmatrix} x_1 \\ y_1 \\ w_1 \end{pmatrix}.$$

Following the same process considering the second camera plane, we obtain:

$$\begin{pmatrix} x_2 \\ y_2 \\ w_2 \end{pmatrix} = [M_{2,in}][M_{2,ex}] \begin{pmatrix} X \\ Y \\ 0 \\ W \end{pmatrix} = M_2 \begin{pmatrix} X \\ Y \\ W \end{pmatrix} = M_2 [M_1]^{-1} \begin{pmatrix} x_1 \\ y_1 \\ w_1 \end{pmatrix} = H \begin{pmatrix} x_1 \\ y_1 \\ w_1 \end{pmatrix}.$$

H is called the homography matrix, and it expresses the relationship between the points in the two image planes of the cameras 1 and 2. As well as M , H is invertible too:

$$\begin{pmatrix} x_1 \\ y_1 \\ w_1 \end{pmatrix} = [H]^{-1} \begin{pmatrix} x_2 \\ y_2 \\ w_2 \end{pmatrix}$$

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

Now a way to compute H is needed. Let consider $w_1 = 1$ and $x_2' = \frac{x_2}{w_2} \wedge y_2' = \frac{y_2}{w_2}$ it is possible to write:

$$\begin{pmatrix} x_2 \\ y_2 \\ w_2 \end{pmatrix} = H \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

$$x_2' = \frac{x_1 * H_{11} + y_1 * H_{12} + H_{13}}{x_1 * H_{31} + y_1 * H_{32} + H_{33}}$$

$$y_2' = \frac{x_1 * H_{21} + y_1 * H_{22} + H_{23}}{x_1 * H_{31} + y_1 * H_{32} + H_{33}}$$

$$x_1 * x_2' * H_{31} + y_1 * x_2' * H_{32} + x_2' * H_{33} - x_1 * H_{11} - y_1 * H_{12} - H_{13} = 0$$

$$x_1 * y_2' * H_{31} + y_2' * H_{32} + y_2' * H_{33} - x_1 * H_{21} - y_1 * H_{22} - H_{23} = 0$$

The homography has nine degrees of freedom. In order to avoid the solution $H = 0^{3 \times 3}$ it is possible to add the constraint $H_{33} = 1$. Now the unknown are eight, and for each pair of points we have two linear independent equations. If it is possible to find four corresponding points, it is possible to compute all the eight homography parameters. The problem written in matrix form is:

$$\begin{bmatrix} -x_{1,1} & -y_{1,1} & -1 & 0 & 0 & 0 & x_{1,1} * x_{2,1}' & y_{1,1} * x_{2,1}' & x_{2,1}' \\ 0 & 0 & 0 & -x_{1,1} & -y_{1,1} & -1 & x_{1,1} * y_{2,1}' & y_{1,1} * y_{2,1}' & y_{2,1}' \\ -x_{1,2} & -y_{1,2} & -1 & 0 & 0 & 0 & x_{1,2} * x_{2,2}' & y_{1,2} * x_{2,2}' & x_{2,2}' \\ 0 & 0 & 0 & -x_{1,2} & -y_{1,2} & -1 & x_{1,2} * y_{2,2}' & y_{1,2} * y_{2,2}' & y_{2,2}' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_{1,n} & -y_{1,n} & -1 & 0 & 0 & 0 & x_{1,n} * x_{2,n}' & y_{1,n} * x_{2,n}' & x_{2,n}' \\ 0 & 0 & 0 & -x_{1,n} & -y_{1,n} & -1 & x_{1,n} * y_{2,n}' & y_{1,n} * y_{2,n}' & y_{2,n}' \end{bmatrix} * \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \\ 1 \end{bmatrix} = 0, (*)$$

The solution to this problem can be easily found using a Singular-Value Decomposition (SVD) approach [Appendix A.1].

2.2 Experimental Homography Matrix Estimation Using Points' Correspondences

In order to have practical implementation of the theoretical research, the previous way of homography estimation has been developed on "Matlab" software.

Two views of the same scene are chosen, a first one constant called "Reference Image" and another one that has been changed during the different tests, called "Moved Image".

The reference image chosen consists of two rectangles, with different areas but the same orientation showed in the picture below.

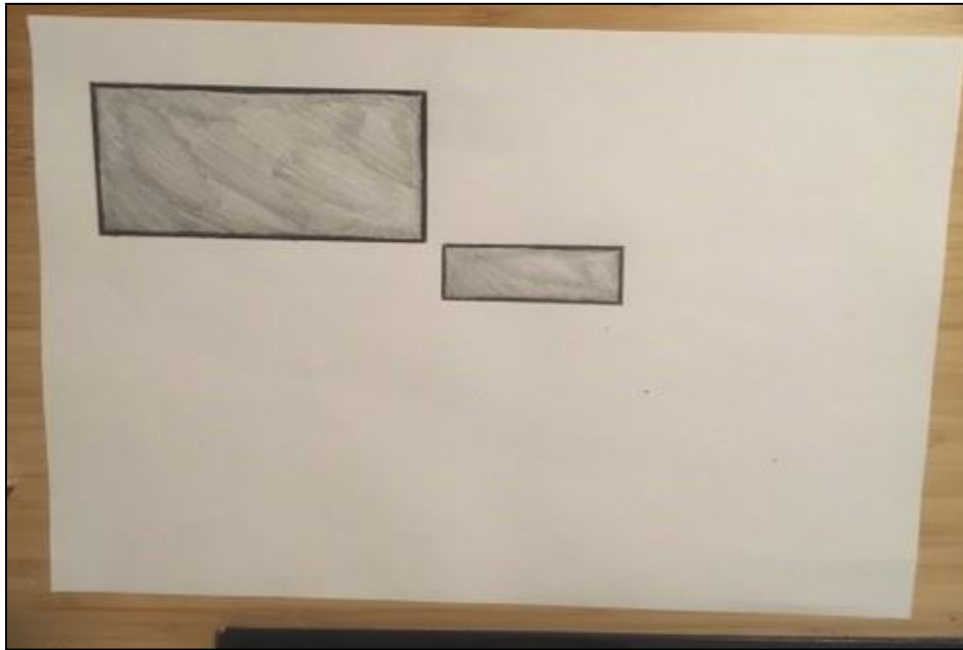


Figure 2.2 - Reference Image used as the first sight in order to estimate the homography matrix.

The algorithm used can be summarized in the following passages:

- Image Processing of the Reference Image;
- Detection of the Rectangles' Perimeters from the Reference Image;
- Identification of the Rectangles' Corners from the Reference Image;
- Image Processing of the Moved Image;
- Detection of the Rectangles' Perimeters from the Moved Image;
- Identification of the Rectangles' Corners;
- Building of the Equations System in Matrix Form (*);
- System Resolution and Homography Estimation using SVD;
- Moved Image Warping;

To find the coordinates of the point in the raster space is fundamental to have a good image processing. First of all the image is uploaded on the software, then it is converted from RGB (true-colour format) to grayscale in order to make it lighter and easier to treat. Now the grayscale is filtered again in order to highlight the different shapes.

After the image has been processed, the corners present are detected using the "Harris Corner Detection" algorithm [Appendix A.2]. From this a further investigation that numerates the corners has been done, based on the geometry of the figure, in order to have the right correspondences.

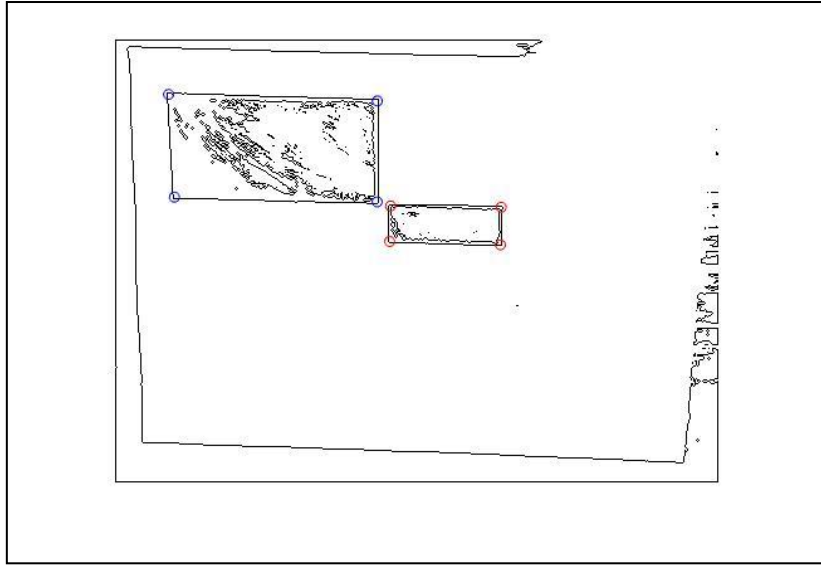


Figure 2.3 - Reference image after being processed. The blue spots are the detected corners of the first rectangle, the red ones are the detected corners of the second rectangle.

This process is done both for the Reference and Moved Image, obtaining the position of the sixteen corners expressed in the raster reference frame, all paired with the correct corresponding one. Now it is possible to compute the homography matrix. This is done through the Singular-Value Decomposition method [Appendix 1], applied to the matrix A :

$$A = \begin{bmatrix} -x_{1,1} & -y_{1,1} & -1 & 0 & 0 & 0 & x_{1,1} * x_{2,1}' & y_{1,1} * x_{2,1}' & x_{2,1}' \\ 0 & 0 & 0 & -x_{1,1} & -y_{1,1} & -1 & x_{1,1} * y_{2,1}' & y_{1,1} * y_{2,1}' & y_{2,1}' \\ -x_{1,2} & -y_{1,2} & -1 & 0 & 0 & 0 & x_{1,2} * x_{2,2}' & y_{1,2} * x_{2,2}' & x_{2,2}' \\ 0 & 0 & 0 & -x_{1,2} & -y_{1,2} & -1 & x_{1,2} * y_{2,2}' & y_{1,2} * y_{2,2}' & y_{2,2}' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_{1,n} & -y_{1,n} & -1 & 0 & 0 & 0 & x_{1,n} * x_{2,n}' & y_{1,n} * x_{2,n}' & x_{2,n}' \\ 0 & 0 & 0 & -x_{1,n} & -y_{1,n} & -1 & x_{1,n} * y_{2,n}' & y_{1,n} * y_{2,n}' & y_{2,n}' \end{bmatrix}, (*)$$

Now that the matrix has been computed it is possible to warp the moved image using H as transformation matrix.

2.3 Tests' Results

The estimator has been tested with different Moved Images and the results are globally good. Below is shown in the picture the results of one test.

In particular are shown the “Moved Image”, the “Reference Image” and the “Warped Moved Image” corresponding to the Moved Image after the warping process. Next to the figure we can see the numerical result of the Homography matrix estimation H .

Moved Image

Reference Image

Warped Moved Image

$H =$

-0.1390	0.9941	0.0100
-0.9932	-0.1466	-0.0023
481.9319	65.0392	1.0000

Test 2.1: As can be seen from the picture, the warping process has produced very good results, with a perfect matching of the two images. The black areas in the “Warped Moved Image” are due to their not visibility in the “Moved Image”.

The several tests taken with different views show that this homography estimation works well.

Nevertheless, this approach is simple but difficult to apply in real situation. Indeed in the real world the presence of rectangular shapes is not negligible at all, and it would be difficult to detect exactly the Reference Image without uncertainty.

A possible solution could be the change of the selected shape, making the detection of the Reference Image easier and more reliable. This solution is discussed in the following chapter.

Chapter 3

Homography Estimation Using Conic Correspondences

The main goal of this research is to implement the homography estimation online on a quadrotor. In the previous chapters, after an introduction to the automation problem for vehicles, the possibility of estimating the homography matrix between two view of the same scene using a set of points' correspondences (at least four) has been discussed. This estimator has been implemented using the corners of two rectangles but, in the real world, the presence of rectangular shape is not negligible, and it could bring some trouble on the shape identification of the reference image. The solution could be finding new shapes in order to avoid uncertainties in the correspondences' identification. This solution has been implemented using conics.

In this chapter it is explained how the homography matrix can be estimated using conics' correspondences, and in particular it is implemented using two ellipsis.

The first section is an introduction to the conics and their numerical representation is developed, followed by a more specific description of the conic of interest, the ellipse.

In the second section, the theoretical way for computing the homography matrix is presented.

In the third section the experimental algorithm, designed by me, used in the experimental tests is explained.

Finally, in the fourth section, the tests' results are discussed.

3.1 Conics: Basic Concepts

3.1.1 Conics

A conic section (or simply a conic) is a curve obtained as an intersection of the surface of a double cone (a cone with two nappes) with a plane. There exists three types of conic:

- the hyperbola;
- the parabola
- the ellipse (circle is a special case of the ellipse);

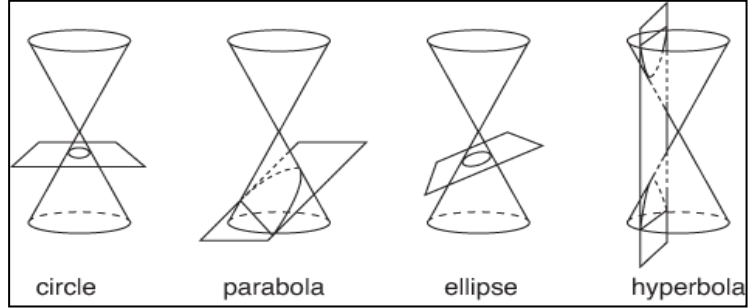


Figure 3.1 - The different types of conic as an intersection of a plane with a double cone.

In analytic geometry, a conic is defined as a planar algebraic curve of degree two, and so a set of points that satisfy a quadratic equation in two variables. If the plane that generate the conic passes through the vertex of the double cone, the intersection generate a point, a line or a pair of intersecting lines; these geometrical figure are called degenerate conics.

A conic section could also be defined as the locus of all points P whose distance from a fixed point F , called focus, is a constant multiple e (called eccentricity) of the distance from P to a fixed line L (called directrix). Studying the eccentricity, we can distinguish the different conics' types:

- $0 < e < 1$ the conic is an ellipse;
- $e = 1$ the conic is a parabola;
- $e > 1$ the conic is an hyperbola;

In the general Cartesian coordinate system, the graph of a quadratic equation in two variables is always a conic section. The most general equation is expressed in the following form:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0.$$

This form could also be represented in matrix notation using homogeneous coordinate as:

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0.$$

Calling C_Q the 3x3 symmetric matrix, it is a very useful way to represent the conics, and is called the matrix of the quadratic. Defining $\mathbf{x} = [x \ y \ 1]$ the homogeneous coordinate, the previous equation could be written as:

$$\mathbf{x}^T C_Q \mathbf{x} = 0.$$

If $\det(C_Q) = 0$ the conic is degenerate. If $\det(C_Q) \neq 0$, computing the minor of the matrix $\det(C_Q) = AC - B^2/4$, it is possible to say:

- if $B^2 - 4AC < 0$ ($\det(C_Q) > 0$), the equation represent an ellipse;
- if $B^2 - 4AC = 0$ ($\det(C_Q) = 0$), the equation represent a parabola;
- if $B^2 - 4AC > 0$ ($\det(C_Q) < 0$), the equation represent a hyperbola;

3.1.2 Ellipsis

An ellipse is a plane curve surrounding two focal points, such that for all points on the curve, the sum of the two distances to the focal points is a constant.

Analytically the equation of a standard ellipse centred around the origin, with width along the x-axis $2a$, and height along the y-axis $2b$ is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

When the centre of the ellipse is shifted to (x_0, y_0) , the equation become:

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1.$$

Another way to represent the ellipse is the standard parametric equation:

$$(x, y) = (a * \cos(t), b * \sin(t)), \text{ for } 0 \leq t \leq 2\pi$$

As stated in the previous paragraph, the conic could be represented by a symmetric 3 x 3 matrix:

$$C = \begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{bmatrix}$$

The coefficients of the this matrix for an ellipse can be computed knowing the semi-major axis a , the semi-minor axis b , the centre coordinates (x_0, y_0) , and the inclination angle θ (angle between the positive horizontal axis and the ellipse's major axis) in the following way:

$$A = a^2(\sin \theta)^2 + b^2(\cos \theta)^2;$$

$$B = 2(b^2 - a^2)\sin \theta \cos \theta;$$

$$C = a^2(\cos \theta)^2 + b^2(\sin \theta)^2;$$

$$D = -2Ax_0 - By_0;$$

$$E = -Bx_0 - 2Cy_0;$$

$$F = Ax_0^2 + Bx_0y_0 + Cy_0^2 - a^2b^2;$$

This representation of the ellipse is crucial in all my work.

3.2 Theoretical Homography Matrix Estimation Using Ellipsis' Correspondences

In order to design an homography matrix estimator I've taken inspiration from the work of Dr. Chum, that finds H defined by first-order Taylor expansion at two (or more) points. A conic C_Q is transformed by a homography transformation H as:

$$\lambda_i C'_Q = H^{-T} C_i H^{-1}.$$

To estimate the homography, the constraints imposed by ellipse-to-ellipse correspondences are studied on the homography through the first order Taylor approximations at two or more different points.

Let $x = (x, y, 1)^T$ be an image point in the first image, $x' = (x', y', 1)^T$ be a point in the second image,

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

be a regular matrix representing a planar homography. We have that $\lambda x' = Hx$, where $\lambda = h_7x + h_8y + h_9$. A first order Taylor expansion $A(H, x)$ of a homography H at a point x is an affine transformation defined as:

$$A = \begin{bmatrix} h_1 - x'h_7 & h_2 - x'h_8 & h_3 + x'(xh_7 - yh_8) \\ h_4 - y'h_7 & h_5 - y'h_8 & h_6 + y'(xh_7 - yh_8) \\ 0 & 0 & h_9 + xh_7 - yh_8 \end{bmatrix}.$$

All homographies that are approximated (first order Taylor approximation) by an affine transformation A at a point x could be expressed as:

$$H = h_7H_7 + h_8H_8 + \lambda A,$$

where

$$H_7 = \begin{bmatrix} x' & 0 & -xx' \\ y' & 0 & -xy' \\ 1 & 0 & -x \end{bmatrix}, H_8 = \begin{bmatrix} 0 & x' & -yx' \\ 0 & y' & -yy' \\ 0 & 1 & -y \end{bmatrix}.$$

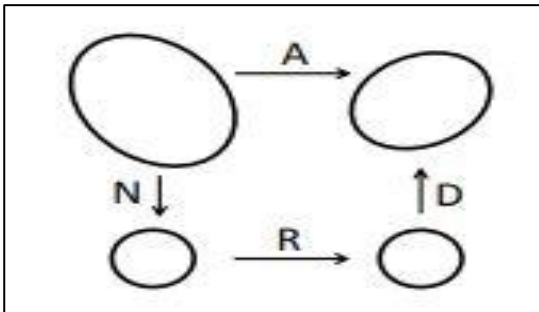


Figure 3.2 - Ellipse-to-ellipse correspondence defines a local affine transformation A up to an unknown rotation R .

If two affine transformations locally approximating the homography at different points are known, equation this equation provides enough linear constraints to estimate the homography H .

A correspondence of two elliptical regions defines an affine transformation up to an unknown rotation

$$A = DRN,$$

where N is an affine transformation normalizing the ellipse in the first image to a unit circle, R is a rotation by an angle α , and D is an affine transformation denormalizing the unit circle into the ellipse in the second image.

Using the notation of $c = \cos \alpha$ and $s = \sin \alpha$. Let $\mathbf{n}_1^T, \mathbf{n}_2^T, \mathbf{n}_3^T = (\mathbf{0}, \mathbf{0}, I)$ be rows of N . A can be rewritten as

$$A = cD \begin{pmatrix} -n_2^T \\ n_1^T \\ 0^T \end{pmatrix} + sD \begin{pmatrix} n_1^T \\ n_2^T \\ 0^T \end{pmatrix} + D \begin{pmatrix} 0^T \\ 0^T \\ n_3^T \end{pmatrix}.$$

Substituting this expression in the equation of the homography we get seven linearly independent equations, linear in 12 unknowns (9 elements of H , λ , λc , and λs). Adding another pair of matching ellipses, seven equations and three new unknowns (λ_i , $\lambda_i c_i$ and $\lambda_i s_i$) are introduced. Therefore, with two pairs of matching ellipses, there are 14 linear equations and 15 unknowns, leading to a one-dimensional space of solutions. Due to the homogeneous nature of the problem, the one dimensional space of solutions for H uniquely determines the homography transformation. For more than two pairs of ellipses, the method leads to a least squares problem.

3.3 Experimental Homography Matrix Estimation Using Ellipsis' Correspondences

The presented method for homography estimation has been implemented on “Matlab” software with the purpose of studying its way of working and its reliability.

The reference image chosen consist of two ellipsis bidimensional, one contained in the other, with perpendicular major axis and different centre, as shown in the figure below.

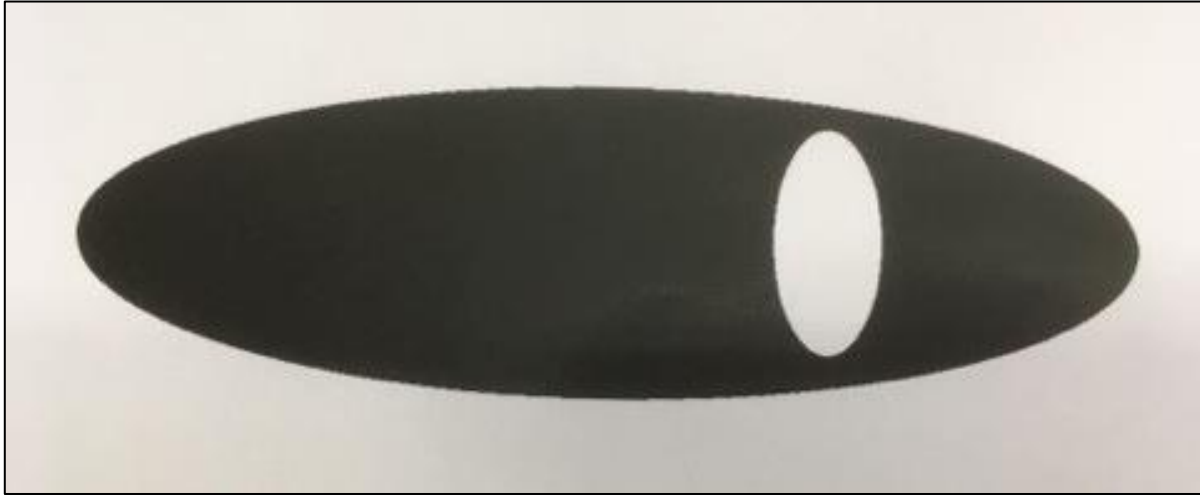


Figure 3.3 - Reference Image used for the homography estimation.

In order to implement the homography estimation the following algorithm has been implemented:

- Image Processing of the Reference Image;
- Extraction of the Main Ellipsis' Parameters from the Reference Image;
- Image Processing of the Moved Image;
- Extraction of the Main Ellipsis' Parameters from the Moved Image;
- Find Homography Matrix using Correspondences of Elliptical Features;
- Warping the Moved Image using the Founded Homography Transformation;
- Resize of the Transformed Image;

In order to estimate the homography, it is fundamental to extract the ellipsis' parameters and order build the quadratic matrix. This is possible analysing all the perimeters present in the scene and choosing the correct ones.

First of all, the image is uploaded and it is converted from RGB (true-colour) to grayscale making the image lighter and easier to treat. Now the grayscale is filtered again, highlighting the different shapes present in the scene. Now all the perimeters present in the image are detected, obtaining as output a binary map (black and white image). A pixel is a part of a perimeter if it is nonzero and is connected to at least one zero-valued pixel.

Now the images are scaled making their measures smaller, in order to have a better estimation that is influenced by the size factor.

The outputs of this initial operation are shown below, both for the Reference and for the Moved Images:

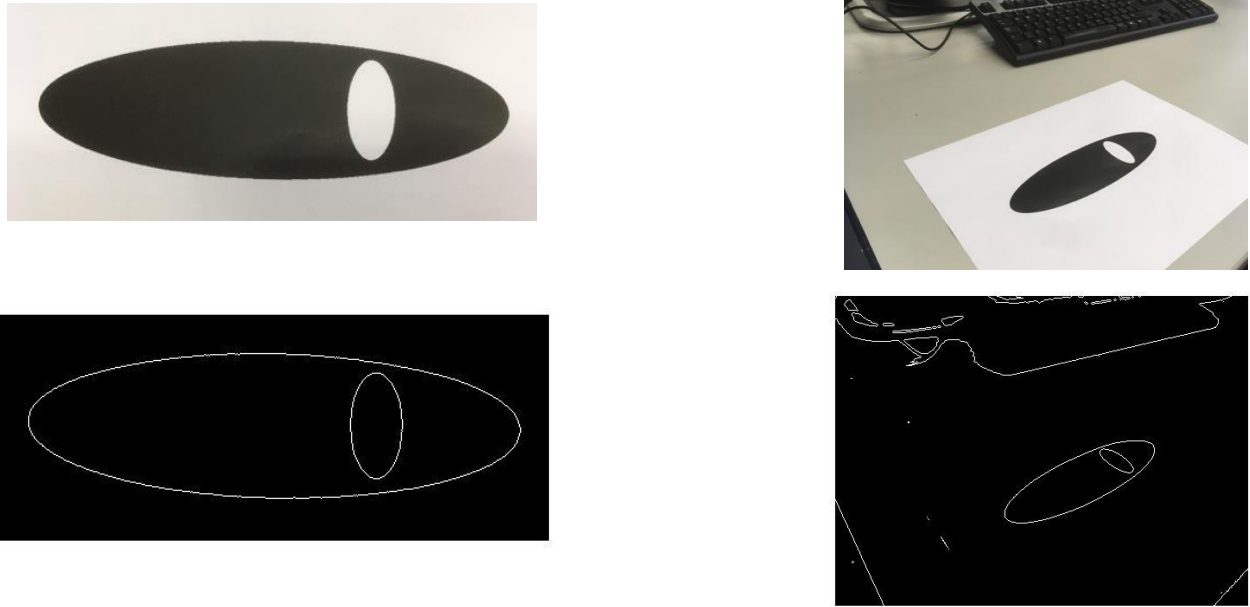


Figure 3.4 - Reference and Moved Image after the Image Processing step.

In order to choose the ellipsis' shapes between all the perimeters detected, I've designed the following algorithm. From the perimeters' map the outer objects (parents) are detected and the ones completely enclosed in them (children). All these boundaries are examined, and the parents that contain only one children, where the latter is empty, is selected as the first ellipse, and the children is the second one.

Now two sets of points are available, corresponding to the ellipsis' perimeters' points. Using these, the best fit to an ellipse is done and the six parameters of the quadratic matrix are computed. In this passage a Least Squares approach is used.

The general equation of an ellipse is:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey = F$$

where $F \neq 0$ and $B^2 - 4AC < 0$. This equation is normalized by dividing both side by F

$$A'x^2 + B'xy + C'y^2 + D'x + E'y = 1,$$

assuming n measurements we define $\mathbf{X} = (x_1, x_2, x_3, \dots, x_n)$ and $\mathbf{Y} = (y_1, y_2, y_3, \dots, y_n)$. Then the following cost function needs to be minimized:

$$C(\beta) = (\mathbf{X}\beta - \mathbf{1})^T (\mathbf{X}\beta - \mathbf{1}),$$

where $\mathbf{X} = [x^2, xy, y^2, x, y]$ is a $n \times 5$ matrix and $\beta = (A', B', C', D', E')^T$ is the parameter's vector to be determined. This problem could be resolved using the least square approach [Appendix A.3] and the solution would be:

$$\beta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{1}$$

The next step is to extract the geometric parameters of the best fitting ellipse from the algebraic equation. If $B \neq 0$ there exist a tilt in the ellipse, so we need to eliminate it. Denoting the tilt angle as θ the following coordinate rotation transformation is employed:

$$\begin{cases} x = \cos\theta x' - \sin\theta y' \\ y = \sin\theta x' + \cos\theta y' \end{cases}$$

after the substitution and some mathematical computation, we obtain the value of θ and a new equation of the non-tilted ellipse in the form:

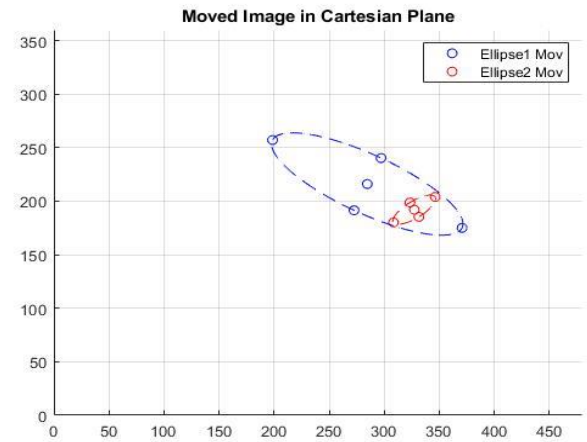
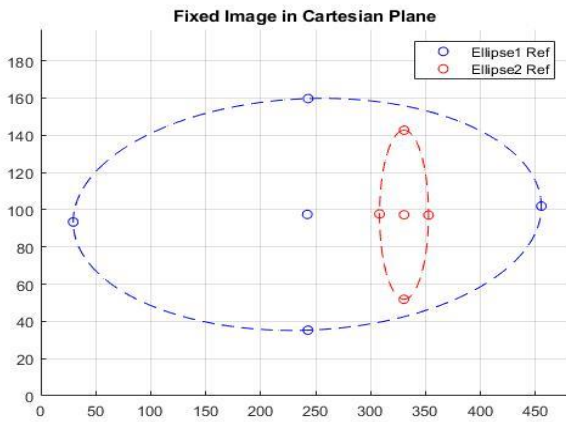
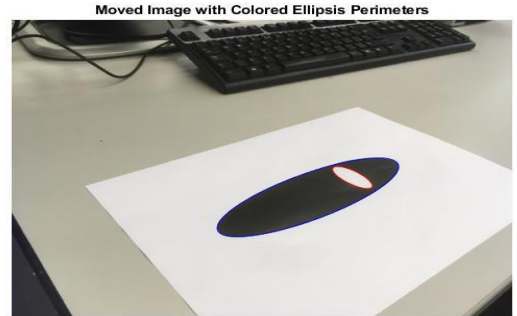
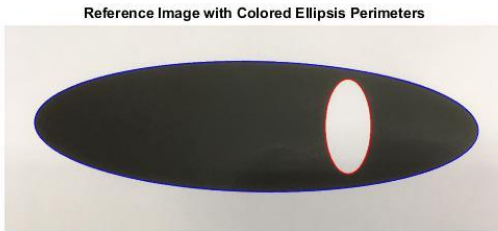
$$A''x'^2 + C''y'^2 + D''x' + E''y' = 1$$

that could be rewritten in the canonical form:

$$\frac{(x' - x'_0)^2}{a^2} + \frac{(y' - y'_0)^2}{b^2} = 1$$

From this new equation it is possible to obtain the non-tilted ellipse's axis length and centre coordinates a , b , x'_0 and y'_0 . From them it is straightforward to obtain the ones of the originals ellipse x_0 and y_0 .

One last passage is done to correct the parameters and to be sure that a is the length of the semi-major axis, b the one of the semi-minor, and that the orientation angle θ is the angle between the semi-major axis and the horizontal one, positive in clockwise direction. Using this parameters (a , b , x_0 and y_0) it is easy to build the matrix representation of the ellipse as described in the previous paragraph. The parameters of the ellipsis can be used in order to plot them in a Cartesian graph.

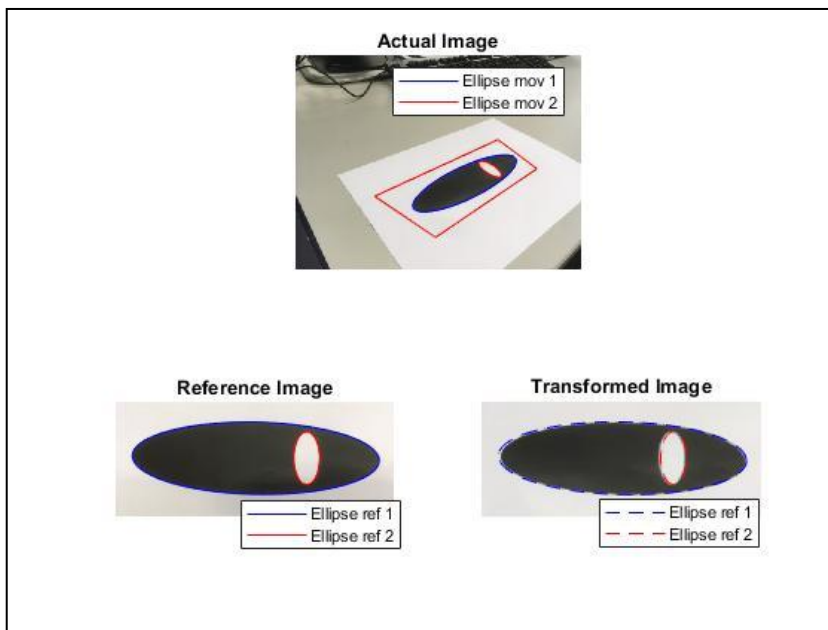


Now that we have the quadratic matrices it is possible to use the estimation presented in the previous chapter. With the estimated homography matrix the Moved Image is warped and transformed.

3.4 Tests' Results

The estimator has been tested with different Moved Images and the results are satisfactory for simple situation, but for extreme ones (Test 3) the transformation is not as good as possible.

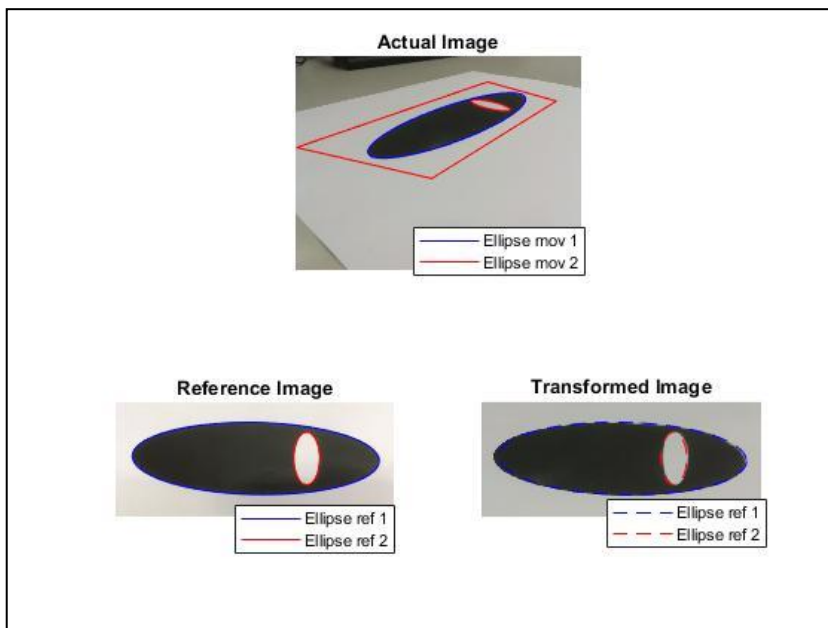
Below some results from the test are presented. In particular are shown the "Actual Image" corresponding to the Moved Image, the "Reference Image" and the "Transformed Image" corresponding to the Moved Image after the warping process. The rectangle present in the "Actual Image" is the boundary of the Reference Image transformed using the inverse of the homography matrix. Next to them it is possible to see the numerical results of the Homography matrix estimation H and a comment.



$$H = \begin{bmatrix} 1.0579 & -1.5714 & 233.6450 \\ 0.8599 & 1.8771 & -559.7337 \\ -0.0012 & 0.0013 & 1.0000 \end{bmatrix}$$

Test 3.1: the Moved Image is rotated and translate with respect to the Reference Image. The optical distortion due to the new point of view are slightly accentuated.

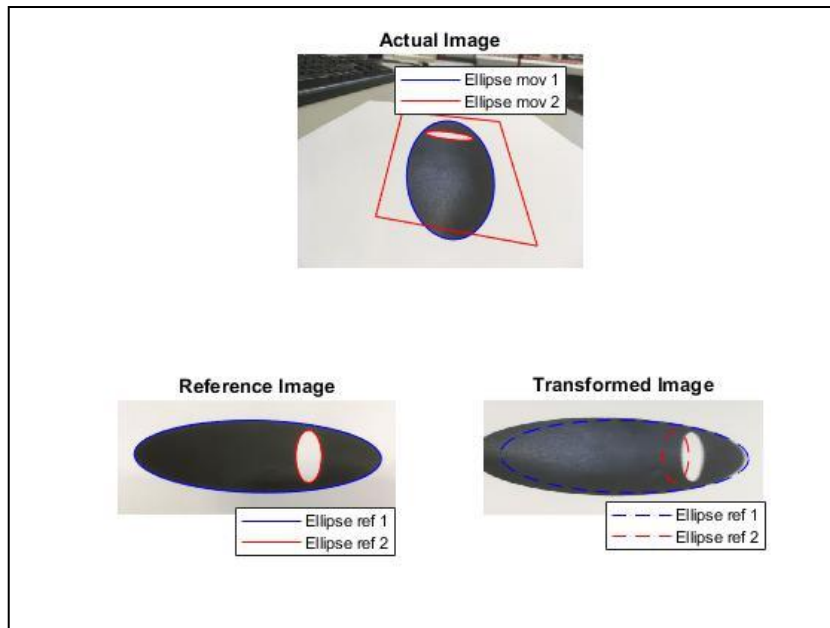
The results of the warping process is very good.



$$H = \begin{bmatrix} 0.2354 & -1.0255 & 316.6292 \\ 0.3575 & 1.0425 & -321.3183 \\ -0.0005 & 0.0015 & 1.0000 \end{bmatrix}$$

Test 3.2: the Moved Image is rotated and translate as in Test 1, but now the optical distortion is more visible.

The results of the warping process are still very good.



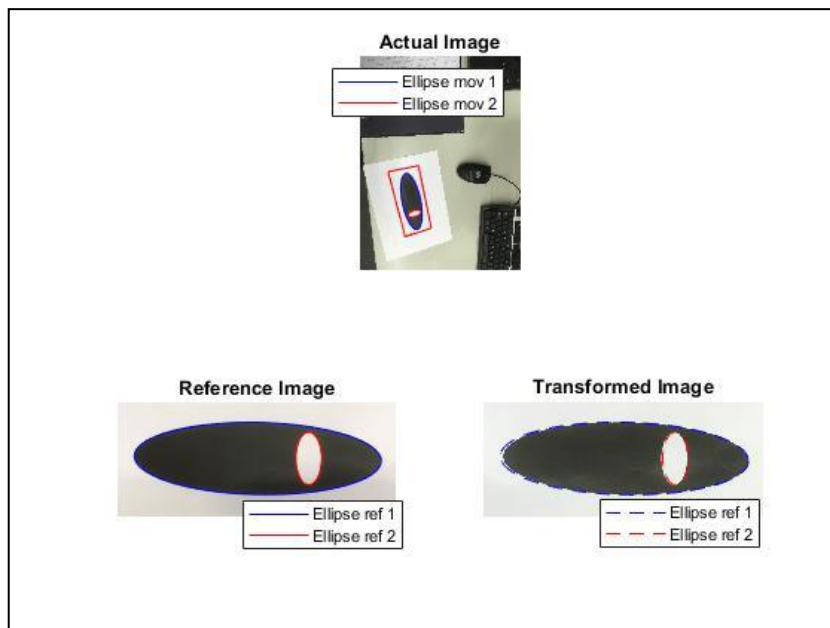
$$H =$$

$$1.0e+03 *$$

0.0007	-0.0040	1.0056
0.0019	0.0005	-0.3907
0.0000	0.0000	0.0010

Test 3.3: the Moved Image is rotated and translate with respect to the Reference Image. The optical distortion now are clearly visible.

The results of the warping process is quite good, it should get better.



$$H =$$

0.3662	-0.0760	-608.6188
0.0749	0.3678	-861.3795
-0.0001	0.0001	1.0000

Test 3.4: the Moved Image is rotated and translate with respect to the Reference Image. Some objects are present in the scene (some with elliptical shape). This test was done to test the searching algorithm.

The results of the warping process is quite perfect.

These tests show that the homography estimator works well in simple situations, especially when the distortion of the shapes due to the perspective is not high. In extreme situations it should be improved. A possible solution could be the implementation of an observer estimator. This solution is discussed in the next chapter.

Chapter 4

Observer Design for Homography Matrix Estimation

In the previous chapters it has been defined what a homography matrix is and what is the way to estimate it, using correspondences between two views of the same planar scene. In particular, first an estimator has been developed that uses as correspondences simple points, and then one that uses as correspondences conic shapes (in our case elliptical shapes). These two methods are a static way of estimation and sometimes they could produce some errors in the results that cannot be avoided.

In this chapter a way of estimating the homography matrix that links two different views that relies on an observer is analysed. The use of an observer is a much reliable method of estimation because, in observability conditions, for $t \rightarrow \infty$, it would produce an error equal to 0. Moreover, the observer would make the computation's time and effort lower, making its implementation on the limited drone's operating system easier.

In the first section a brief introduction to the state observer is done.

In the second section the theory behind the observer design is explained, describing the algorithm used for its implementation.

Finally, in the third section the tests' results is shown and commented.

4.1 Introduction to State Observer

In control theory, a state observer is a system that provides an estimate of the internal state of a given real system, from measurements of the input and output of the real system. It is typically computer-implemented, and provides the basis of many practical applications.

The problem of homography estimation has been extensively studied in the previous chapters. However, that works consider the homography as an incidental variable and are not focused on improving the estimation over time. The quality of the homography estimates depends heavily on the nature of the image features exploited as well as the algorithm used.

One of the objectives of this new approach to develop a feature-based homography estimation algorithm that increase the quality of the estimation over time rather than computing algebraically individual raw homography for each image. This approach is useful also in the next chapters because it would exploits the temporal correlation of data across a video sequence.

The methodology taken exploits the underlying structure of the Special Linear group $SL(3)$, a Lie group isomorphic to the group of homographies. This property has, however, seldom been considered in observer design for smoothing the homography estimates.

The observer has the capacity of encompassing static and non-static cases. In this chapter is presented in the tests' results only the static cases.

If applied to a video, the observer would achieve the temporal smoothing of noisy data, and providing good homography estimates even in situations where the number and quality of conic-feature correspondences on a frame-by-frame basis are insufficient to directly compute raw homographies, during a short time period.

4.2 Nonlinear Observer Design on the Special Linear group $SL(3)$

In order to design the observer, I have taken inspiration from the work of Dr. Hua “Explicit Complementary Observer Design on Special Linear Group $SL(3)$ for Homography Estimation using Conic Correspondences”.

- Observer Design

The Special Linear group $SL(3)$ is defined as:

$$SL(3) = \{H \in \mathbb{R}^{3 \times 3} | \det H = 1\},$$

and its Lie algebra is:

$$sl(3) = \{U \in \mathbb{R}^{3 \times 3} | \text{tr}(U) = 0\}.$$

A conic is expressed in homogeneous with a matrix $C \in \mathbb{R}^{3 \times 3}$, and if is non-degenerate its determinant is not null. Since the relation $x^T C_Q x = 0$ implies $x^T (\lambda C_Q) x = 0$ for $\lambda \neq 0$, all λC_Q represent the same conic. Therefore, without loss of generality, it can be assumed that the conic matrix belongs to the set defined as:

$$\mathbb{C} = \{C_Q \in \mathbb{R}^{3 \times 3} | C_Q^T = C_Q, \det(C_Q) = 1\}.$$

The homography matrix belongs to $SL(3)$, and its kinematics is expressed as:

$$\dot{H} = F(H, U) = HU,$$

where $U \in sl(3)$ is the group velocity of the camera.

We consider $C'_k \in \mathbb{C}$ the set of conic measurements and $C_k \in \mathbb{C}$ the reference conic matrices, constant and known. Measurements are expressed in the camera current frame:

$$C'_k = h(H, C_k) = H^T C_k H, k = \{1, 2\}.$$

Let \hat{H} denote the estimate of H , we could define the group error $E := \hat{H}H^{-1} \in SL(3)$, so that \hat{H} converge to H if and only if E converges to identity. Moreover, we define the output errors as:

$$e_k = \rho(\hat{H}^{-1}, C'_k) = \hat{H}^{-T} C'_k \hat{H}^{-1} = E^{-T} C_k E^{-1},$$

where ρ is a right group action. The proposed observer in Hua et al. paper takes the form:

$$\dot{\hat{H}} = \hat{H}U - \Delta(\hat{H}, C'_k)\hat{H},$$

where $\Delta(\hat{H}, C) \in sl(3)$ is the innovation term to be designed. If we find Δ right invariant, the dynamics of the group error E are autonomous and given by:

$$\dot{E} = -\Delta(E, C_Q)E,$$

In order to determine (\hat{H}, C) , a non-degenerate right-invariant cost function is needed. We first define individual degenerate right-invariant cost at C_k as:

$$C_{c_k}(\hat{H}, C'_k) = \frac{1}{2} \|\hat{H}^{-T} C'_k \hat{H}^{-1} - C_k\|_{K_k}^2,$$

with $K_k \in \mathbb{R}^{3 \times 3}$ symmetric positive definite matrix. Once verified that $C_{c_k}(\hat{H}, C'_k)$ are right invariant, the aggregate cost function defined as the sum of all individual costs is also right invariant.

Using a right invariant Riemannian metric on $SL(3)$, the cost function is treated such that the innovation term results:

$$\Delta(\hat{H}, C_Q) = -\mathbb{P} \left(\sum_{k=1}^N e_k(e_k - C_k)K_k + e_k K_k(e_k - C_k) \right),$$

Where \mathbb{P} is the projection operator:

$$\mathbb{P}(A) = A - \frac{1}{3} \text{tr}(A)I.$$

Using this formulation is possible to demonstrate that $\Delta(E, C) = \Delta(\hat{H}, C)$, and subsequently:

$$\dot{E} = -\mathbb{P} \left(\sum_{k=1}^N e_k(e_k - C_k)K_k + e_k K_k(e_k - C_k) \right) E,$$

if

$$\bigcap_{k=1}^N \ker(d\rho_{C_k}(I)) = \{0\}.$$

Is satisfied the equilibrium $E = I$ of System \dot{E} is locally exponentially stable.

- Observability Analysis

In order to satisfy the algebraic constraint and thus the local observability, the following lemma is used:

Lemma (Local Observability) Assume there exist two non-degenerate conics C_1, C_2 belonging to the set of all observed conic such that $M = C_1 C_2^{-1}$ has three distinct non-null eigenvalues. Then the algebraic constraint:

$$\bigcap_{k=1}^N \ker(d\rho_{C_k}(I)) = \{0\}.$$

Is satisfied and the equilibrium $E = I$ is locally stable.

This is true when the matrix $M = C_1 C_2^{-1}$ has three distinct and real eigenvalues.

- Discrete Version

A discrete version could be implemented using the accelerated line search (ALS) algorithm. This method is based on update formula:

$$x_{k+1} = R_{xk}(t_k \eta_k),$$

where R is a retraction. The simplest approach to optimize a differentiable function is to translate $x(t)$ in the direction of the steepest descendent ($-\text{grad}(f)$) until the gradient vanishes (stationary point). t_k is the length obtained from the Armiko back-tracking procedure.

For the proposed observer, considering $U \ll 0$ ($U \approx 0$), the update formula at each iteration step is given by:

$$\hat{H}_{k+1} = e^{-\Delta_k t_k} \hat{H}_k,$$

and t_k relies on the following Armijo-like condition:

$$C_Q(\hat{H}_k, C') - C_Q(\hat{H}_{k+1}, C') \geq t_k^A \sigma \|\Delta_k\|_F^2, t_k^A = \beta^m \bar{\alpha},$$

where m is the smallest nonnegative integer that satisfy the former condition. The chosen parameters are $\bar{\alpha} = 0.05$, $\beta = 0.75$, and $K_k = \text{diag}([1, 1, 2])$.

- System Description

The system can be seen as a non-linear system and can be described by a state-space representation. Considering the following parameters:

- H : homography matrix from the reference image to the moved one;
- C_1, C_2 : conic matrices in the reference image (constant and known);
- C'_1, C'_2 : conic matrices in the moved image;
- U : group velocity of the homogeneous matrix;

The system is represented with the following formulation:

- State (x) = H ;
- Input (u) = U ;
- Output (y) = C'_1, C'_2 ;

The state-space representation would be:

$$\dot{H} = F(H, U) = HU$$

$$C'_k = h(H, C_k) = H^T C_k H, k = \{1, 2\}$$

The observer using the input (U) and the output of the system (C'_k) should be able to compute the state (H). Its formulation is:

$$\dot{\hat{H}} = F(U, C') = \hat{H}U - \Delta(\hat{H}, C')\hat{H}.$$

Designing $\Delta(\hat{H}, C')$ the stability and the observability of this system could be controlled as described in this paragraph.

A continuous time implementation has been done using Simulink. Below could be seen the block diagram used.

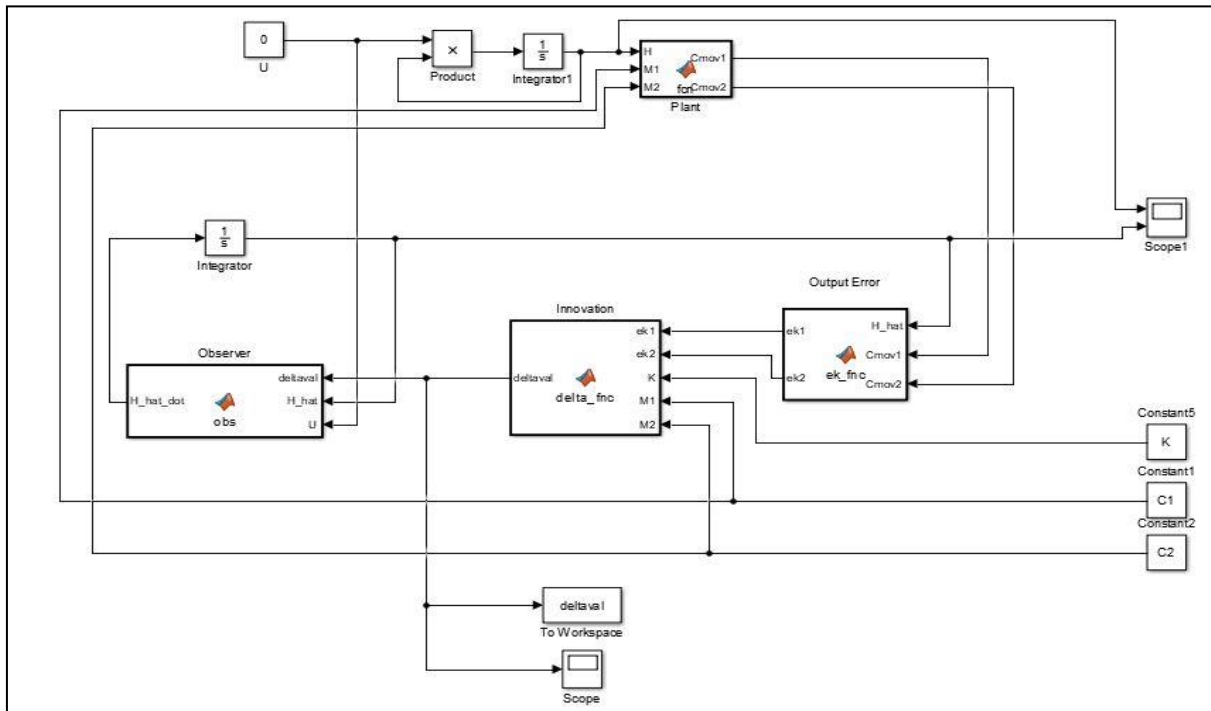


Figure 4.1 - Block Diagram of the Simulink implementation of the observer's continuous version.

- **Algorithm Used for Discrete Time Implementation**

The algorithm used is similar to the one proposed for the estimation using the Taylor approximation of the homography matrix, described in Chapter 2.

In order to implement the homography estimation the following passages has been implemented:

- Image Processing of the Reference Image;
- Extraction of the Main Ellipsis' Parameters from the Reference Image;
- Image Processing of the Moved Image;
- Extraction of the Main Ellipsis' Parameters from the Moved Image;
- Find Homography Matrix using Discrete Version of the Observer Estimator;
- Warping the Moved Image using the Last Homography Matrix Estimation;
- Resize of the Transformed Images;

The observer works better when the homography matrix's coefficients to be estimated are smaller, for this reason the pictures taken from the camera are treated in order to decrease the value of the conic matrices' coefficient. In particular, before computing the conic matrix, I first make of the centre making it coinciding with the origin of the pixel map, and a scaling of 1:300 is applied. After the warping process the inverse transformation are applied.

In order to implement the discrete version of the observer, he chosen parameters are $\bar{\alpha} = 0.05$, $\beta = 0.75$, and $K_k = \text{diag}(11, 1, 21)$.

Moreover, for the choice of the first value of the estimation $H_hat(:, :, 1)$, different approaches have been used. Initially a random matrix that satisfy the constraint $H \in SL(3)$ have been used. After these attempts, the observer has been feed with an initial estimation equal to the H static estimation, computed through the method explained in Chapter 2.

In the following paragraph are presented the results with $k = 1000$.

4.3 Tests' Results

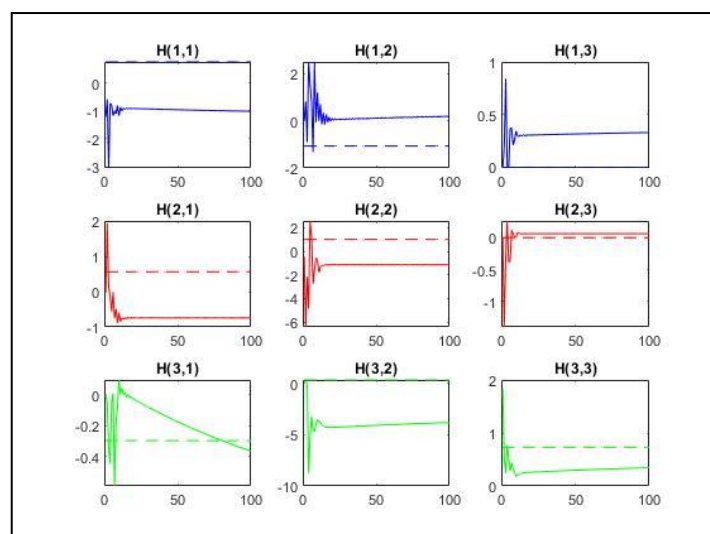
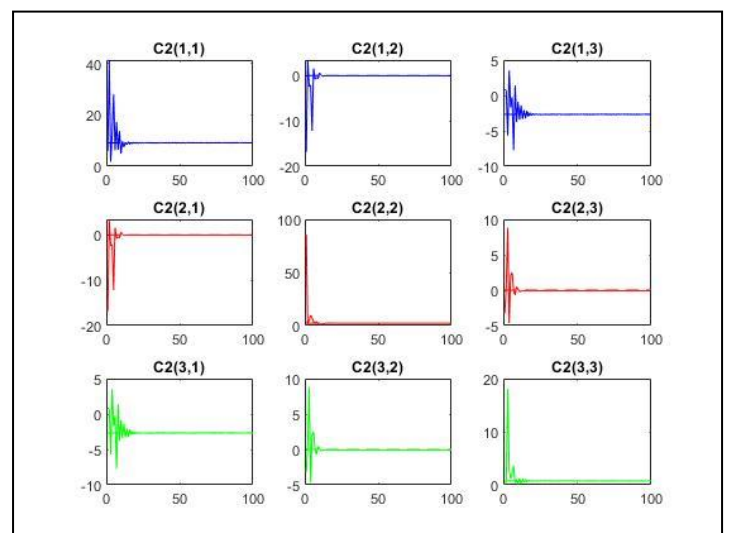
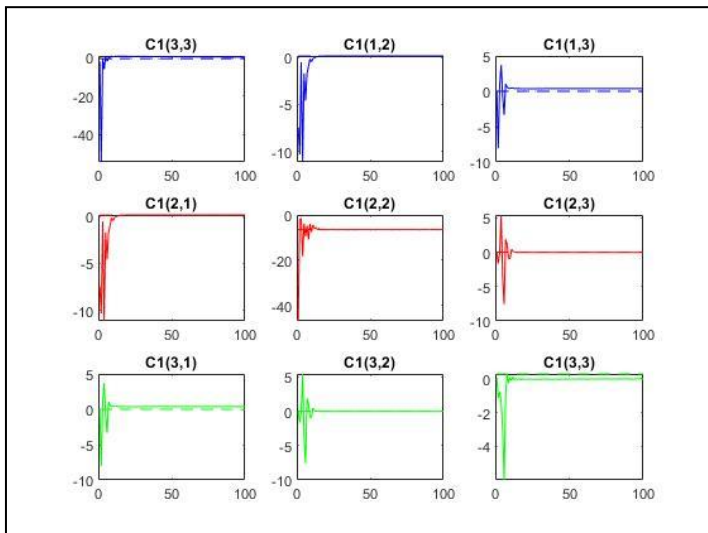
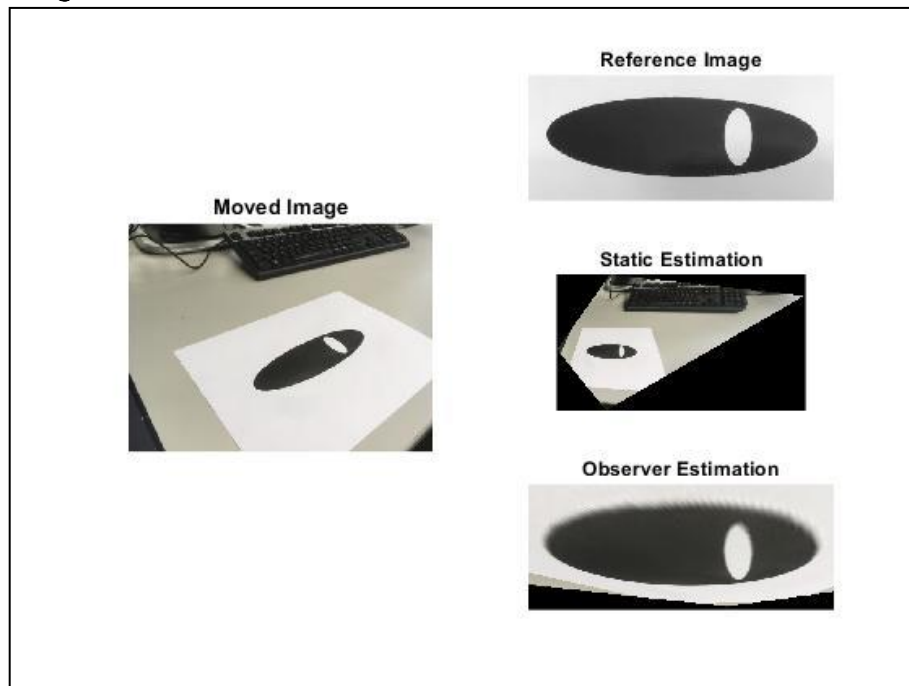
The estimator has been tested with different moved Images and with different initial estimate value of the homography matrix. The results are satisfying; they are good with initial random H and increase the estimation quality with initial static estimation H .

In the following pages are presented the main tests. The moved Images are the same used in the previous chapter. The first four cases are done using as initial value of \hat{H} a random diagonal matrix belonging to $SO(3)$. The second ones are done using the static estimation. The observer has worked for $k = 1000$ discrete time steps.

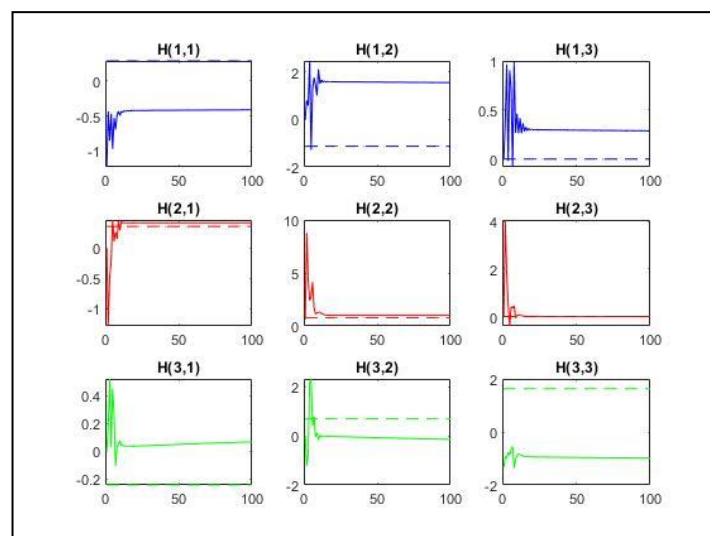
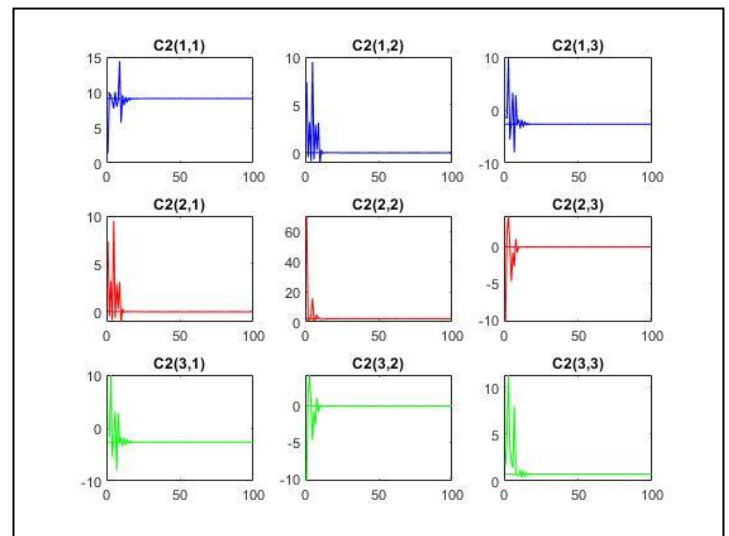
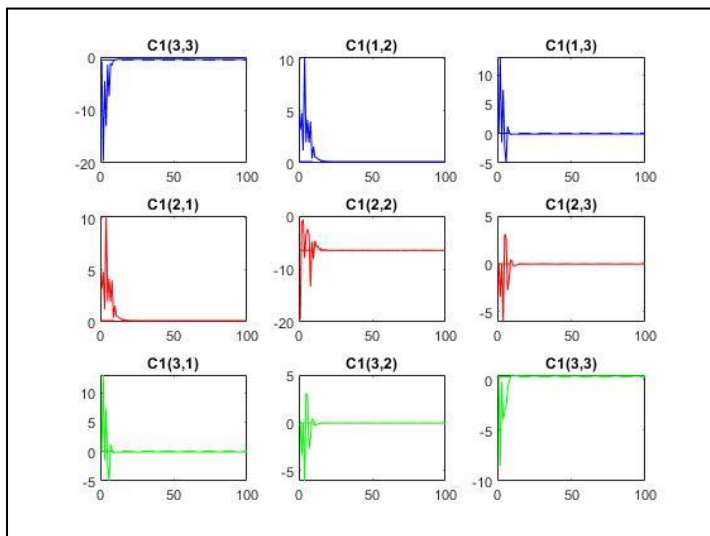
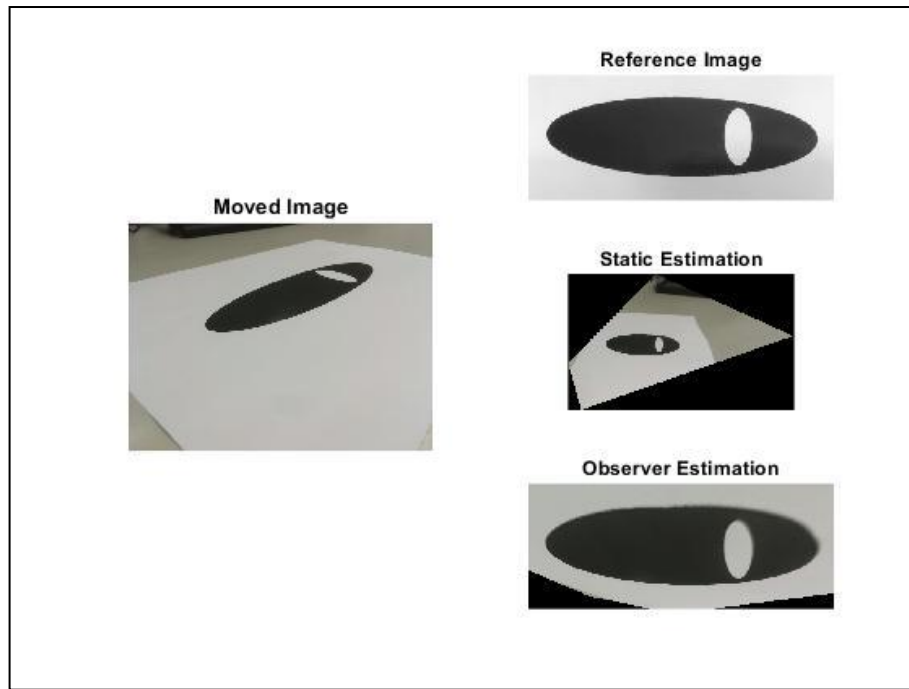
In the presentation of the results are inserted also the graph of $C_Q(\hat{H}_k, C')$ and \hat{H} variation. In particular the continues line are the estimated matrices' parameters while the dashed lines are the value of the same parameters computed using the static method. In order to appreciate in a more clear way the step variation of these parameters, they are presented only for the first $k = 100$ discrete time step.

The observability of the different cases has been checked before each test.

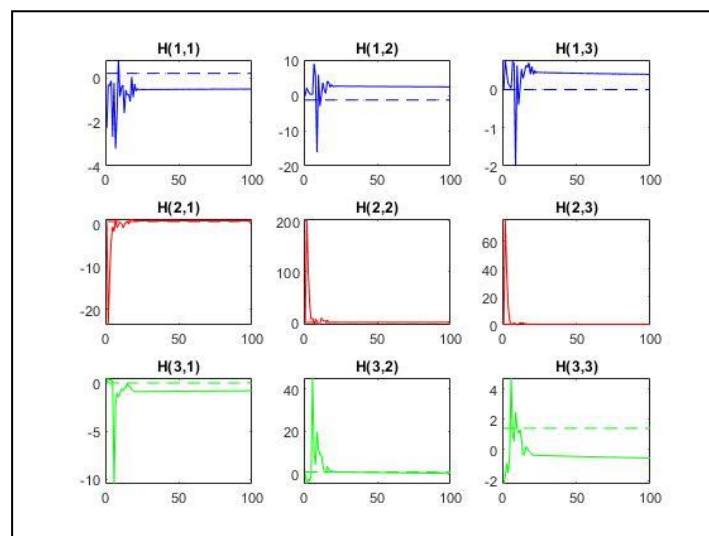
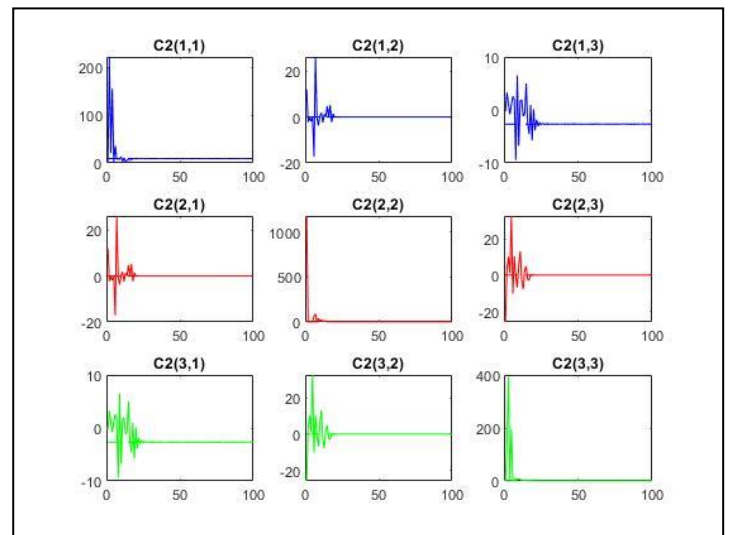
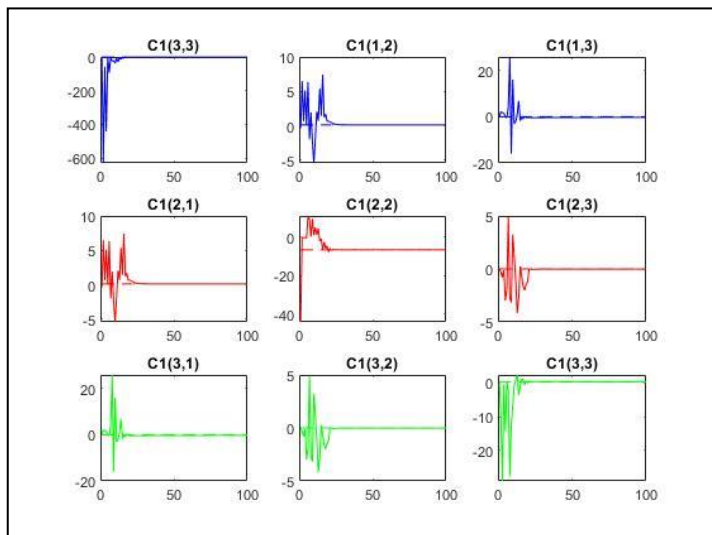
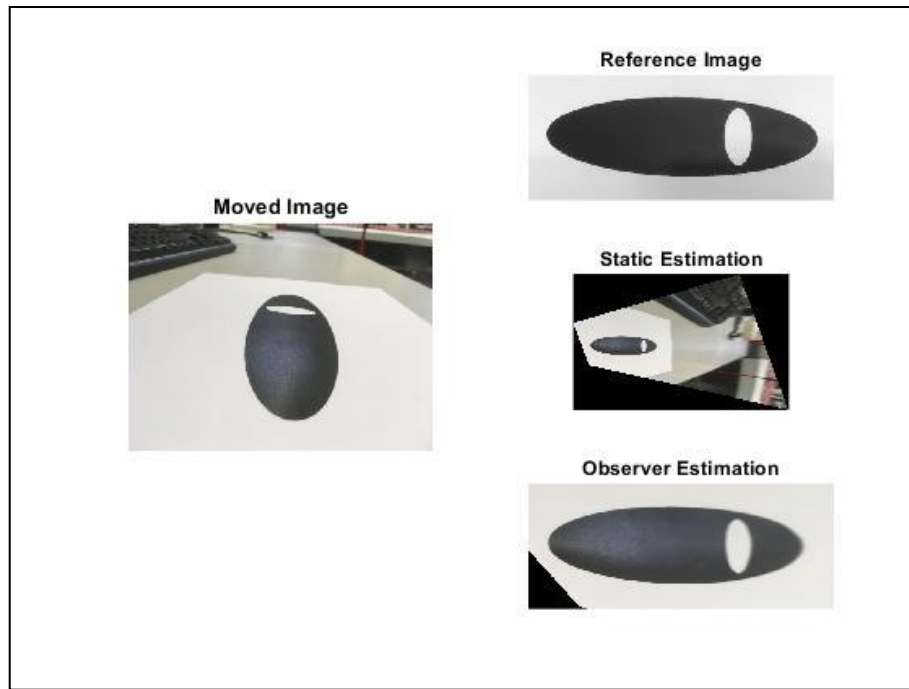
- Tests using random matrix as initial value of \hat{H}



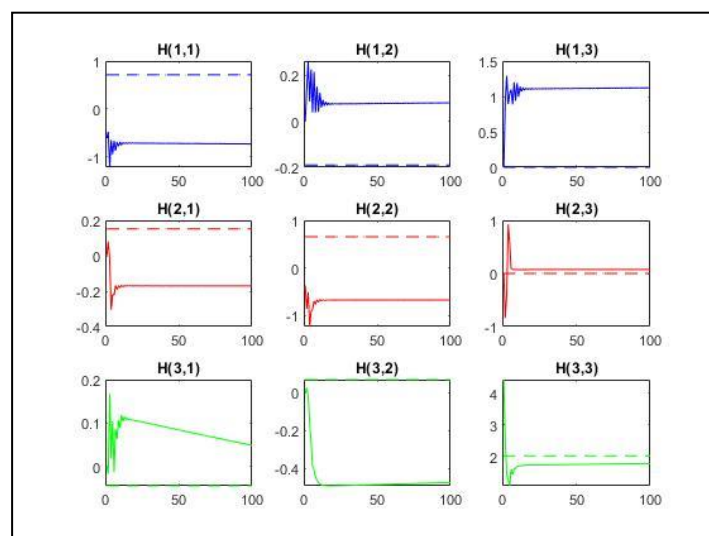
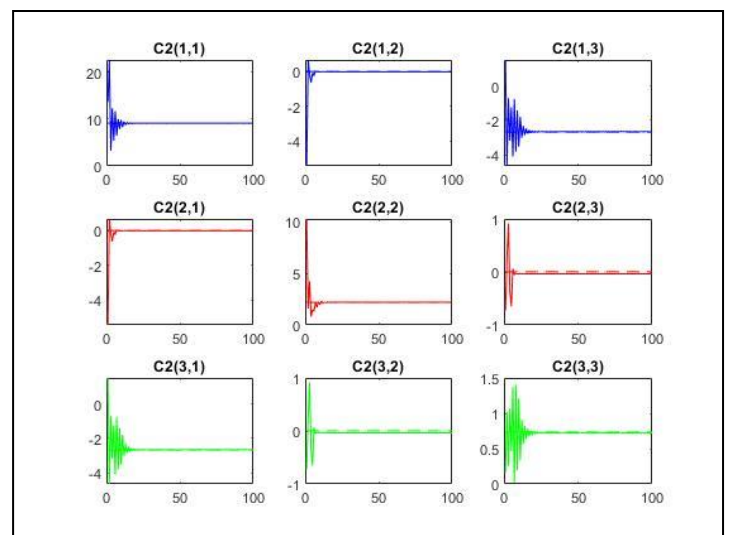
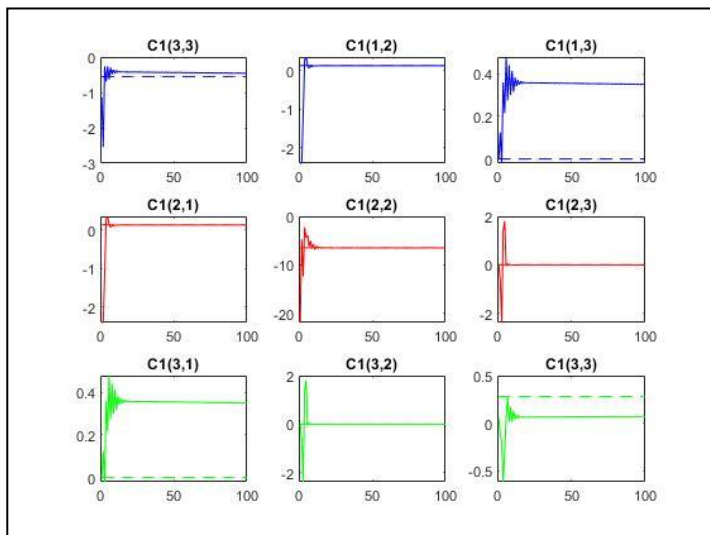
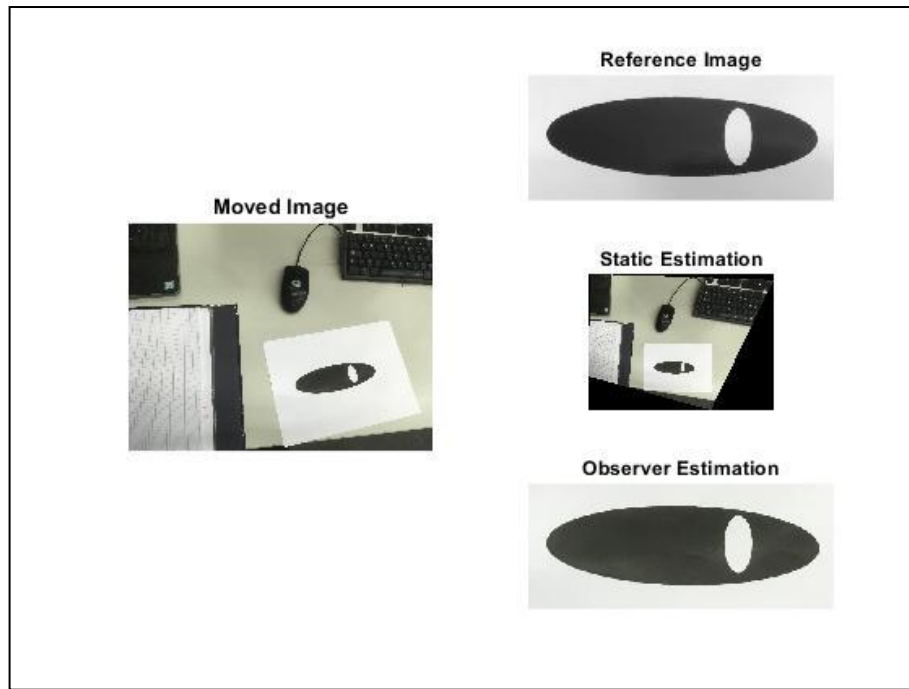
Test 4.1: using random matrix as initial value of \hat{H}



Test 4.2: using random matrix as initial value of \hat{H}

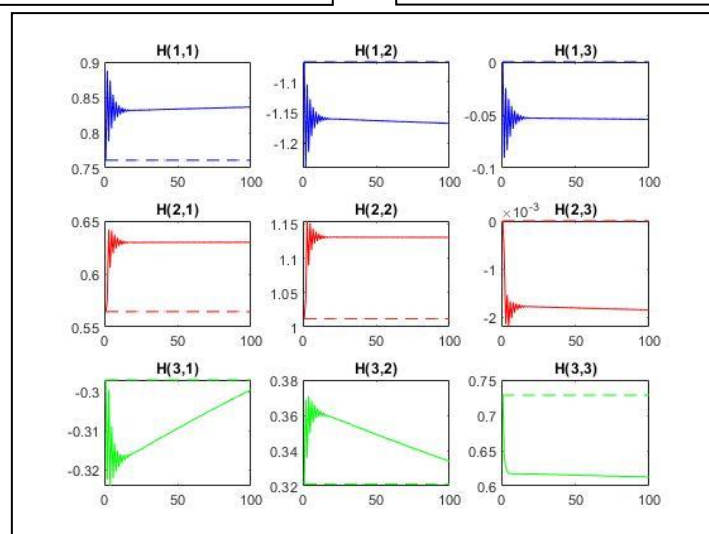
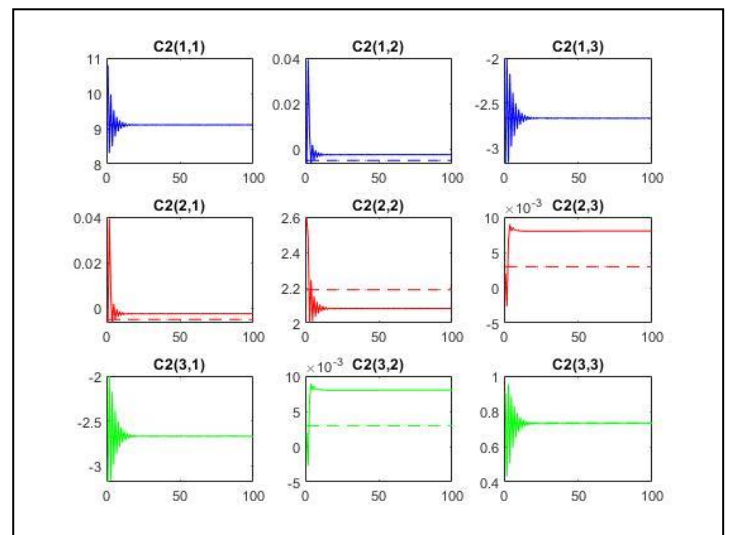
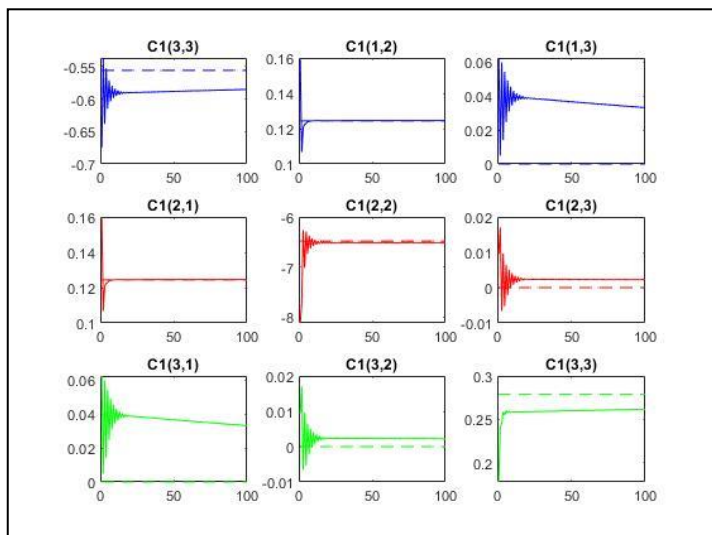
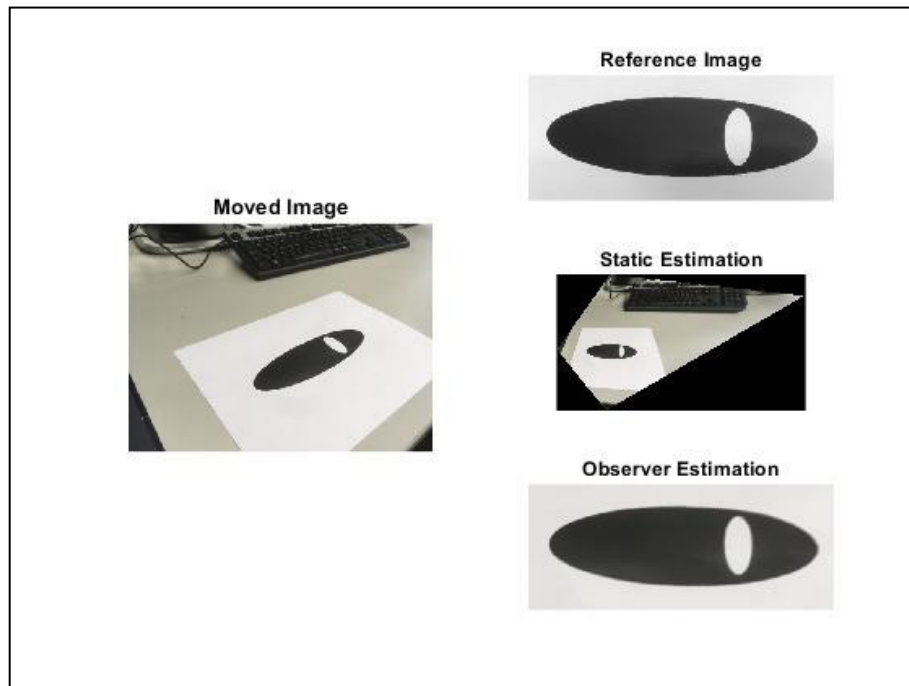


Test 4.3: using random matrix as initial value of \hat{H}

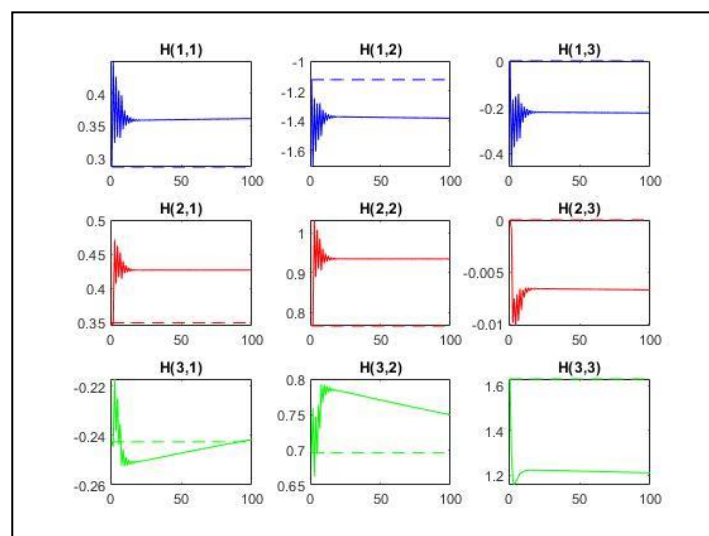
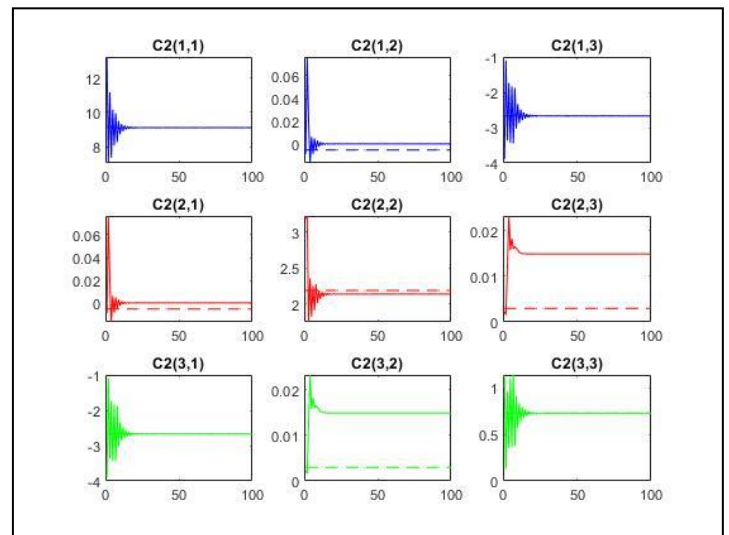
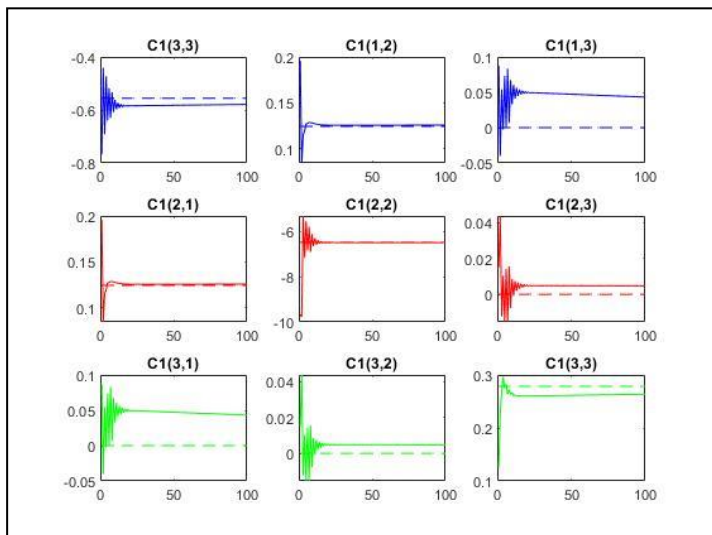
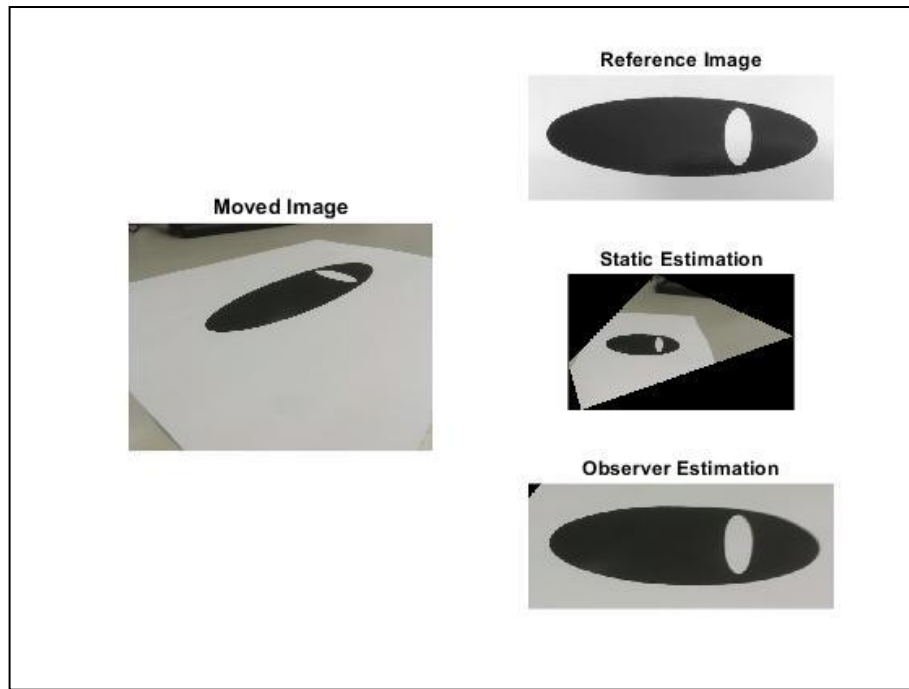


Test 4.4: using random matrix as initial value of \hat{H}

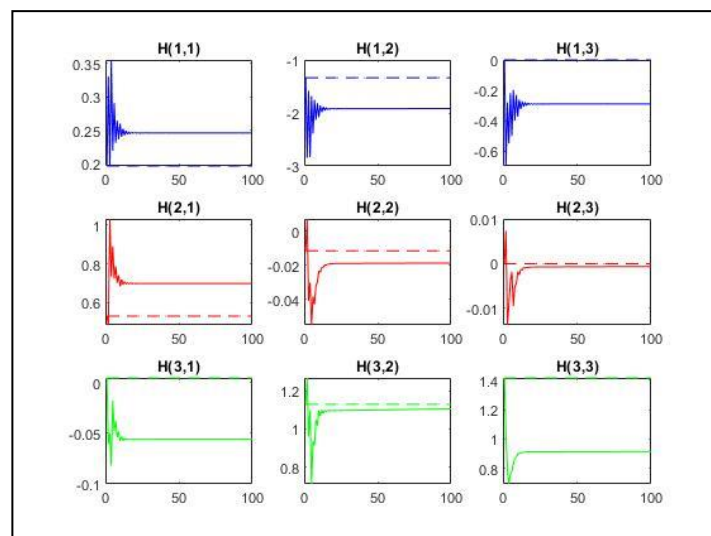
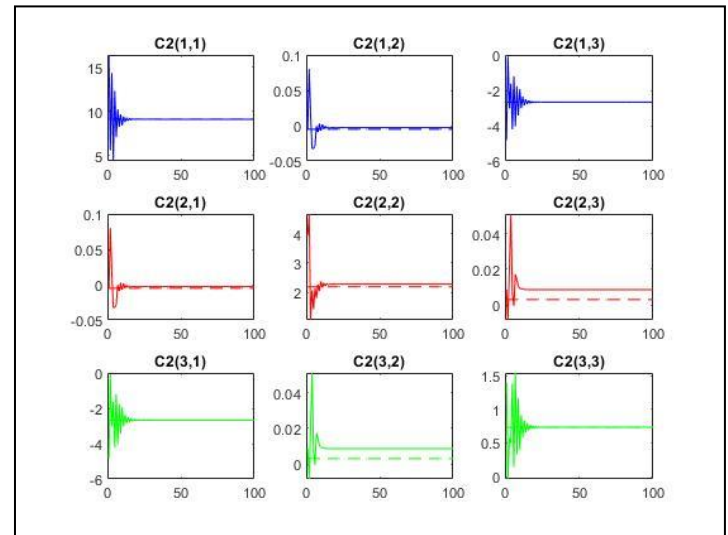
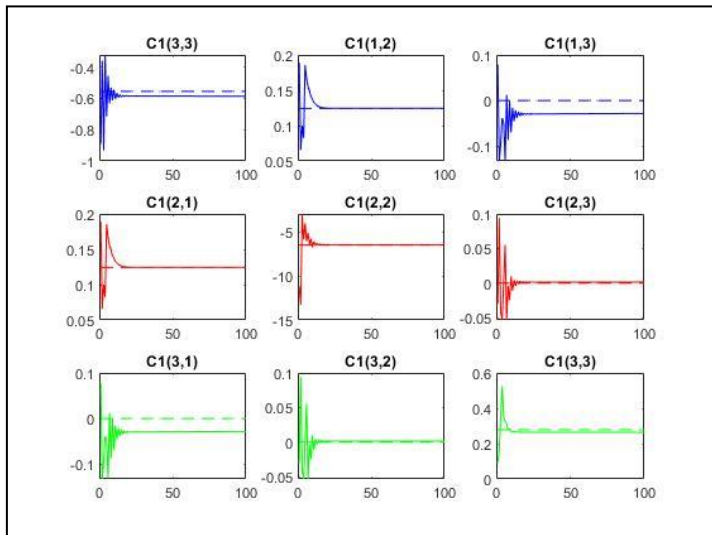
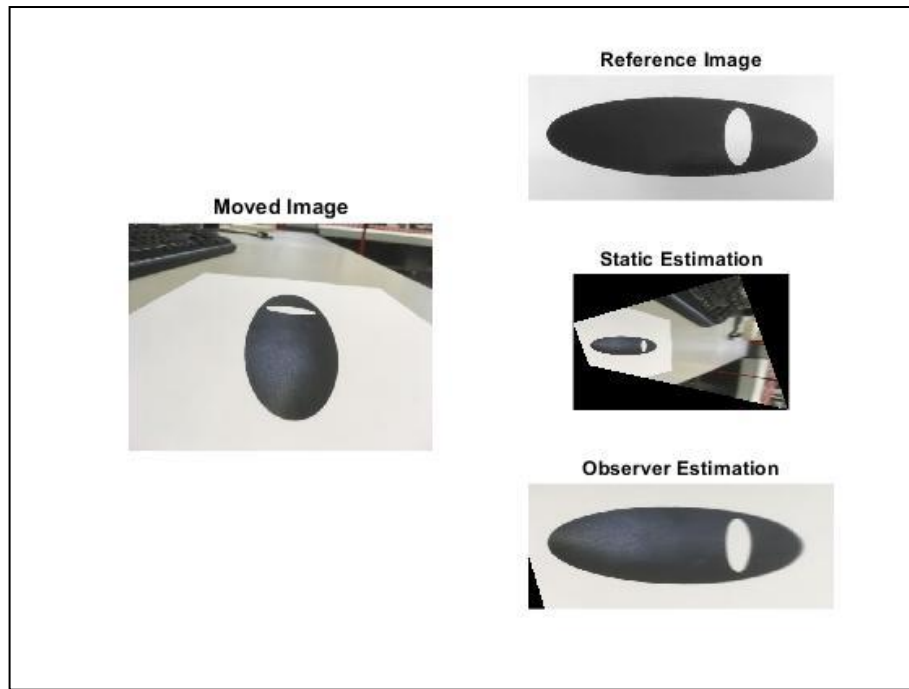
- Tests using static estimation of H as initial value of \hat{H}



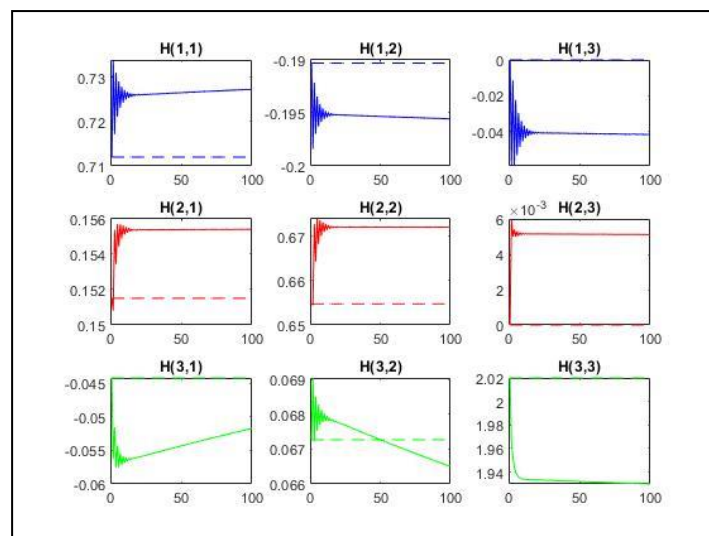
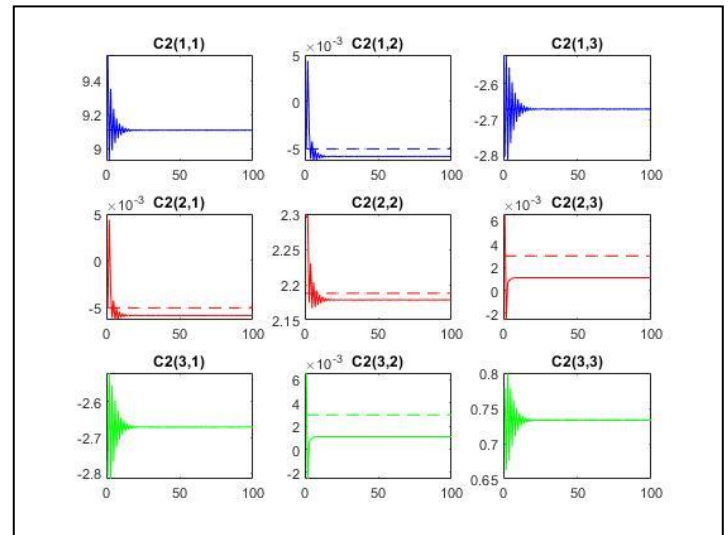
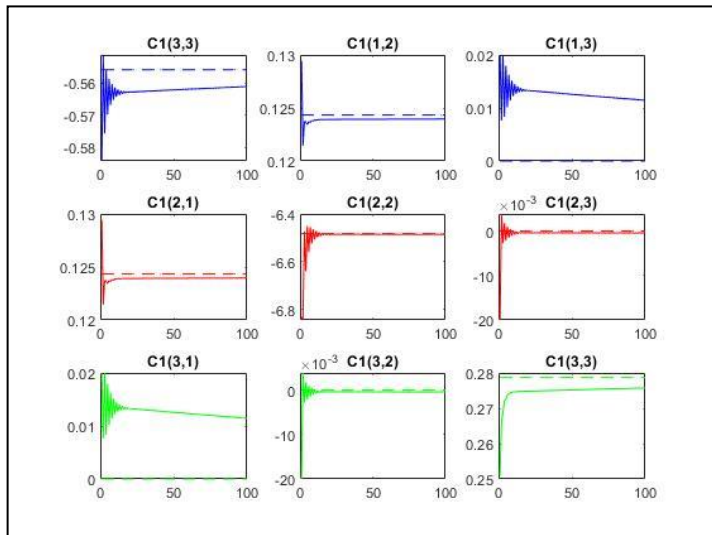
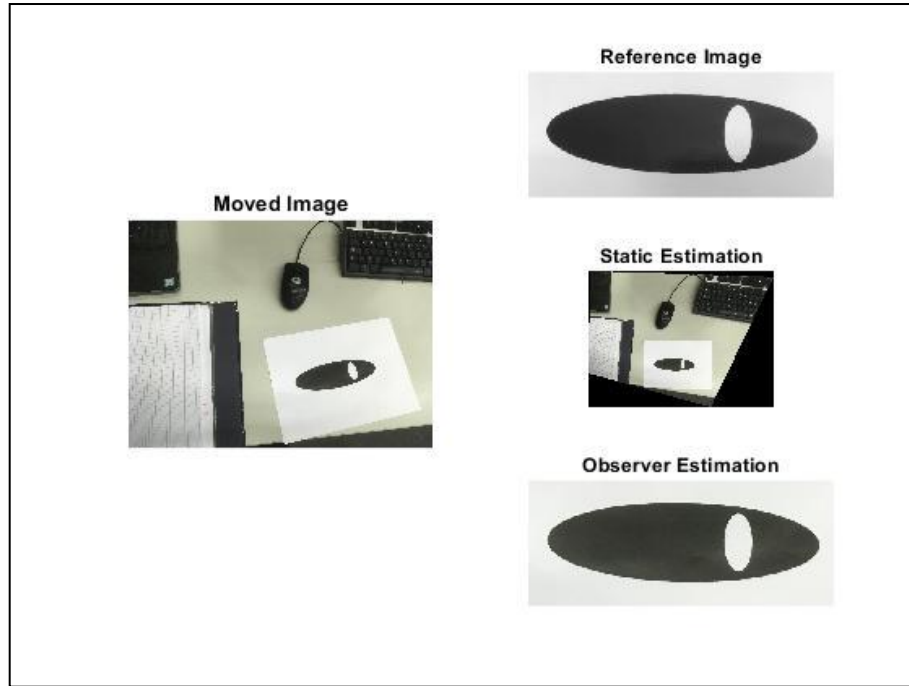
Test 4.5: using static estimation of H as initial value of \hat{H}



Test 4.6: using static estimation of H as initial value of \hat{H}



Test 4.7: using static estimation of H as initial value of \hat{H}



Test 4.8: using static estimation of H as initial value of \hat{H}

These tests show that the observer for homography estimation works well. If we have any information about the homography matrix (first four cases) the estimation converges bringing the conic's quadratic matrix estimated to converge toward the original ones. If we have some information (last four cases) the homography estimation is improved making the final results better.

In the following chapter the different way of estimation designed is implemented on a video in order to study their way of working for a sequence of frame linked to each other.

Chapter 5

Video Rectification using Homography Estimation

In the previous chapters different ways of estimating the homography matrix have been designed and simulated using the software “Matlab”. These estimators have been tested on a static case, where the Moved Images are still. Remembering that the final goal of this research is to implement the homography estimation on the drone’s operating system, this estimator should be tested in a non-static situation, where the Moved Image changes in time.

In this chapter it is explained how the Static Estimator (designed in Chapter 3) and the Observer Estimator (designed in Chapter 4) have been used in order to estimate the homography matrix between a Reference Image and a Video. After this process, an image rectification is done.

In the first section, the algorithm for video rectification using the Static Estimator is explained, showing the results for the main representative frames.

In the second section, the same procedure is done using the Observer Estimator, showing the main results for this case.

The last section reports the comparison of the two methods’ results.

5.1 Video Implementation of Static Estimation

The estimation of the homography matrix using conic correspondence is described in depth in the Chapter 3. In this paragraph, an application of this algorithm to video image stabilization is presented. The experimentations have been performed with a classical webcam that provides 30 images per second with a resolution of 3840x2160 pixels.

The algorithm used could be summarized in the following passages:

- Image Processing of the Reference Image;
- Extraction of the Main Ellipsis' Parameters from the Reference Image;
- Video Processing;
- Static Estimator Application to Every Frames;

A video is nothing more than a sequence of images, called frames. The uploaded video is so segmented in a sequence of frames. For every single frames, the Algorithm designed in Chapter 3 has been used.

Here follow the results for some frames of interest. The Moved Images is the instantaneous frames. The Reference Image is the same used in the previous chapters.

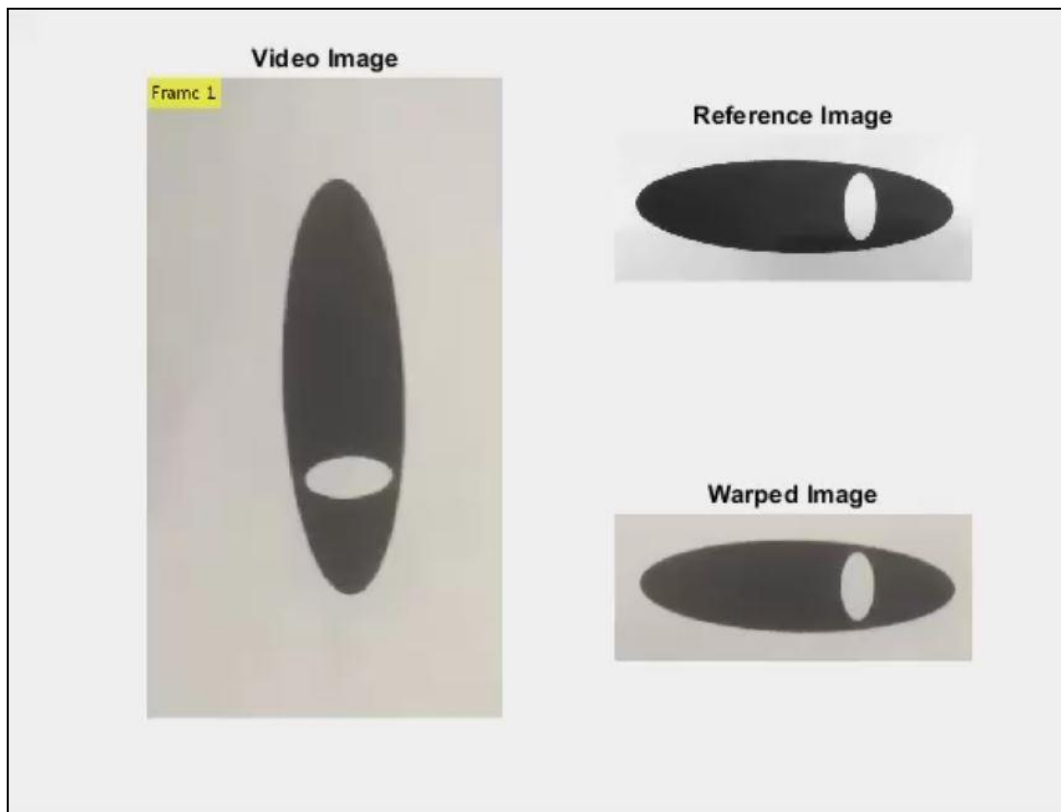


Figure 5.1 - Frame 1 rectification done after using the Static Estimator.

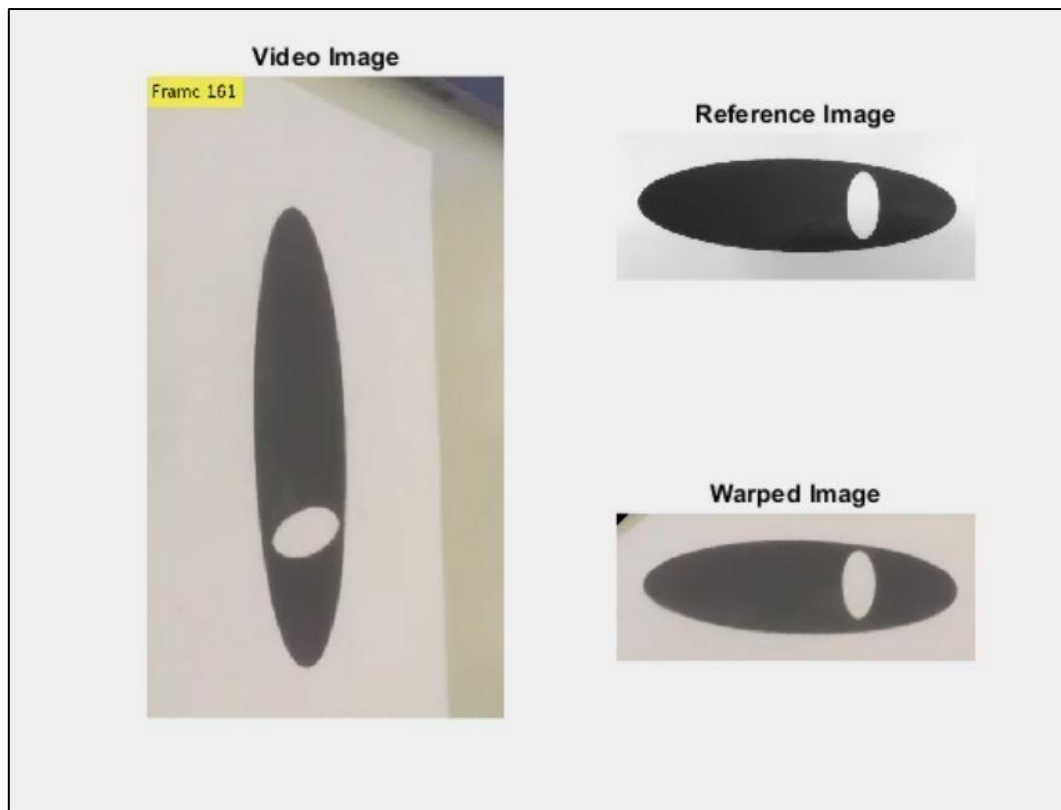


Image 5.2 - Frame 161 rectification done after using the Static Estimator.

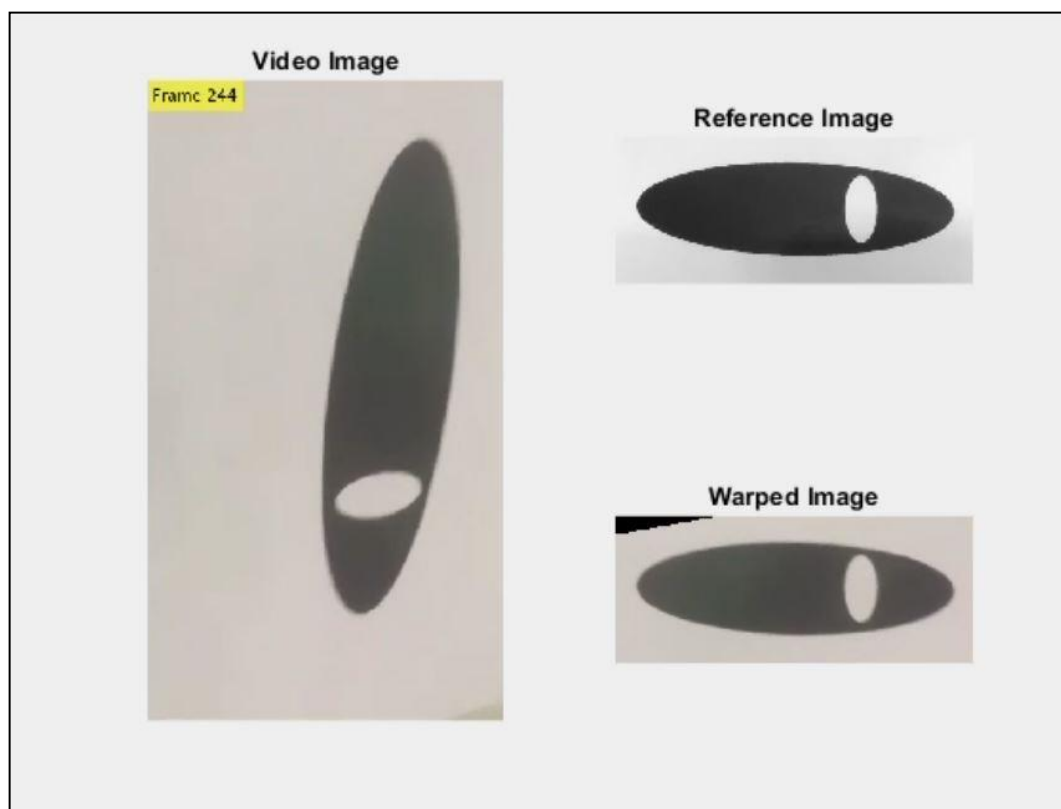


Figure 5.3 - rame 244 rectification done after using the Static Estimator.

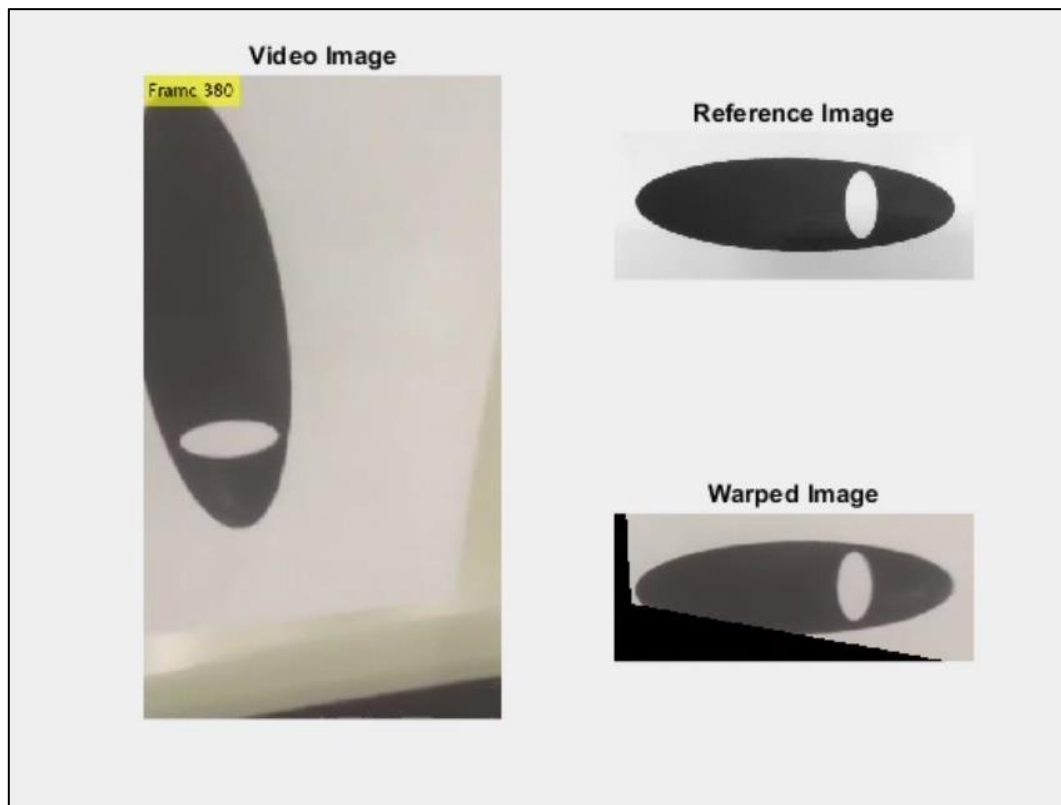


Figure 5.4 - Frame 380 rectification done after using the Static Estimator.

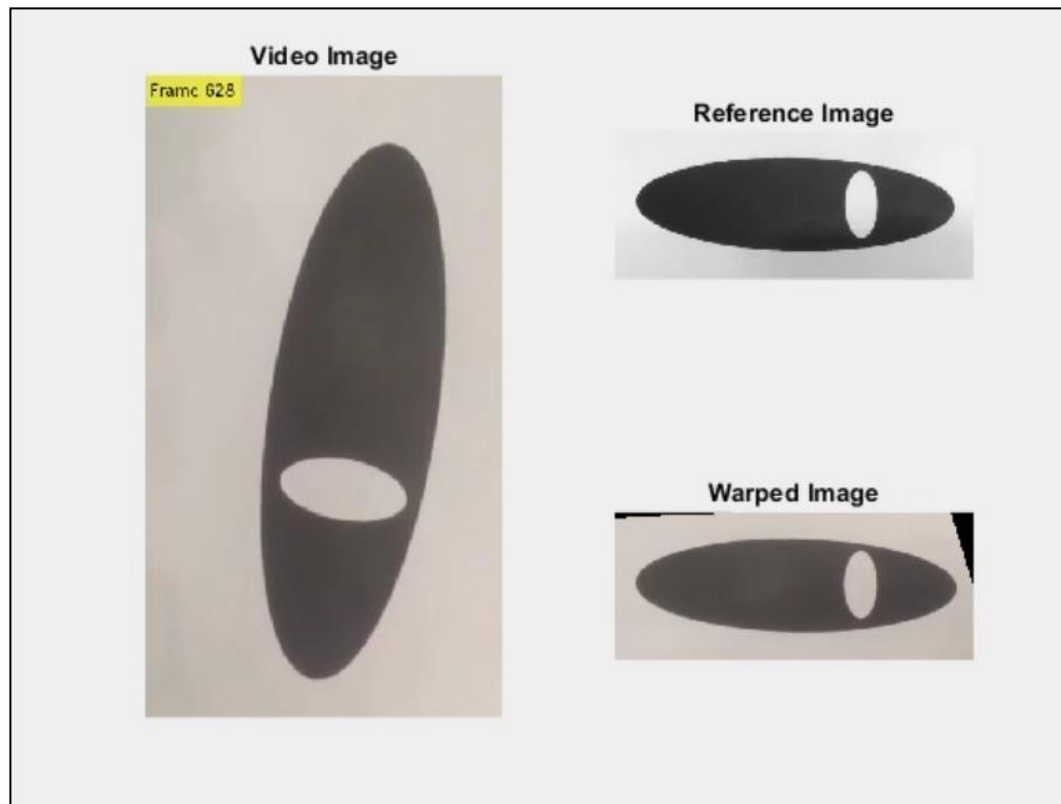


Figure 5.5 - Frame 628 rectification done after using the Static Estimator.

5.2 Video Implementation of Observer Estimation

The estimation of the homography matrix using a state observer is described in depth in the Chapter 4. In this paragraph, an application of this algorithm to video image stabilization is presented. The experimentations have been performed with a classical webcam that provides 30 images per second with a resolution of 3840x2160 pixels.

In order to have better performance for the image rectification process, the homography matrix using the first frame is estimated with the Static Estimator, then the Observer is used from the second frame. The first value of the estimated homography matrix is set to the final value of the previous frame's estimation. The number of discrete steps used in each frames is $k = 1000$.

The algorithm used could be summarized in the following passages:

- Image Processing of the Reference Image;
- Extraction of the Main Ellipsis' Parameters from the Reference Image;
- Video Processing;
- Static Estimator Application to the First Frame;
- Observer Estimator Application to all the other Frames;

Here follow the results for some frames of interest. The Moved Images is the instantaneous frames. The Reference Image is the same used in the previous chapters.

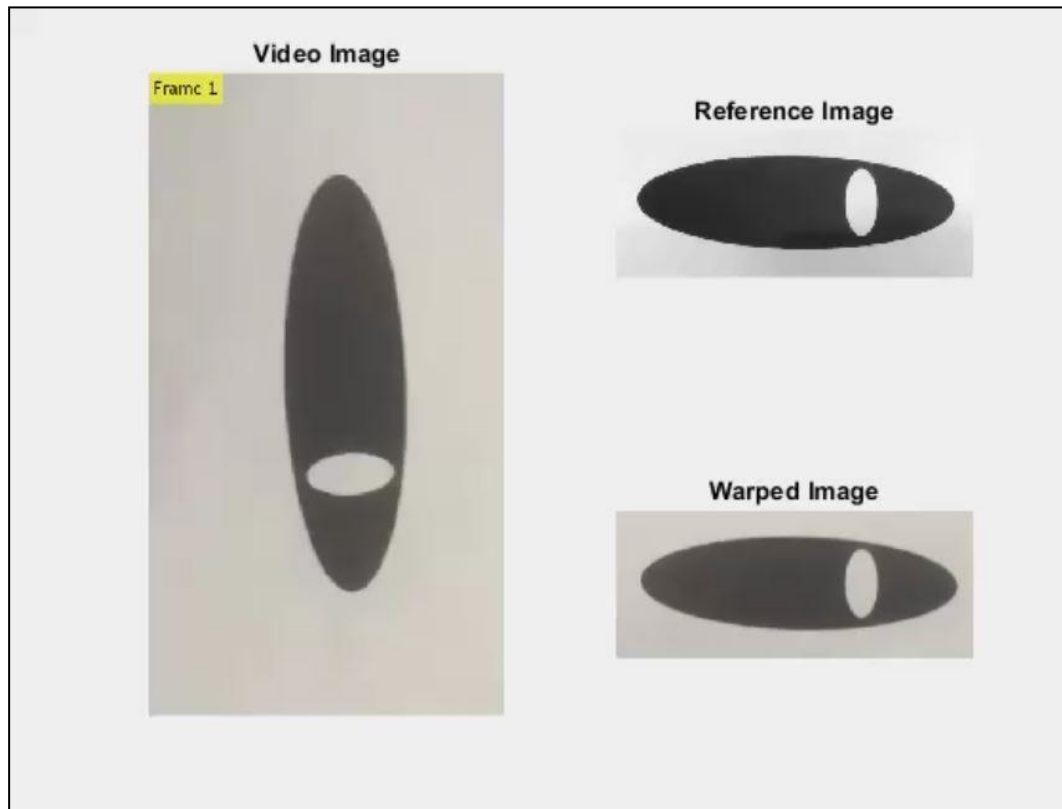


Figure 5.6 - Frame 1 rectification done after using the Static Estimator. This estimator is used only for the first frame.

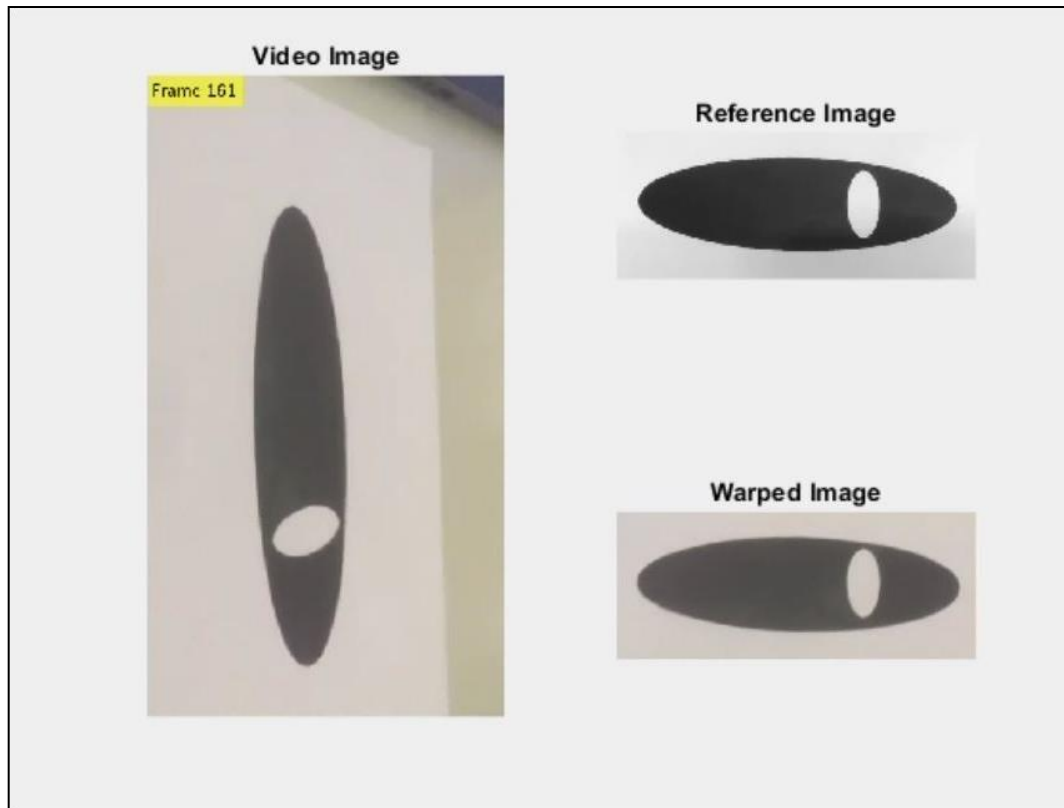


Figure 5.7 - Frame 161 rectification done after using the Observer Estimator.

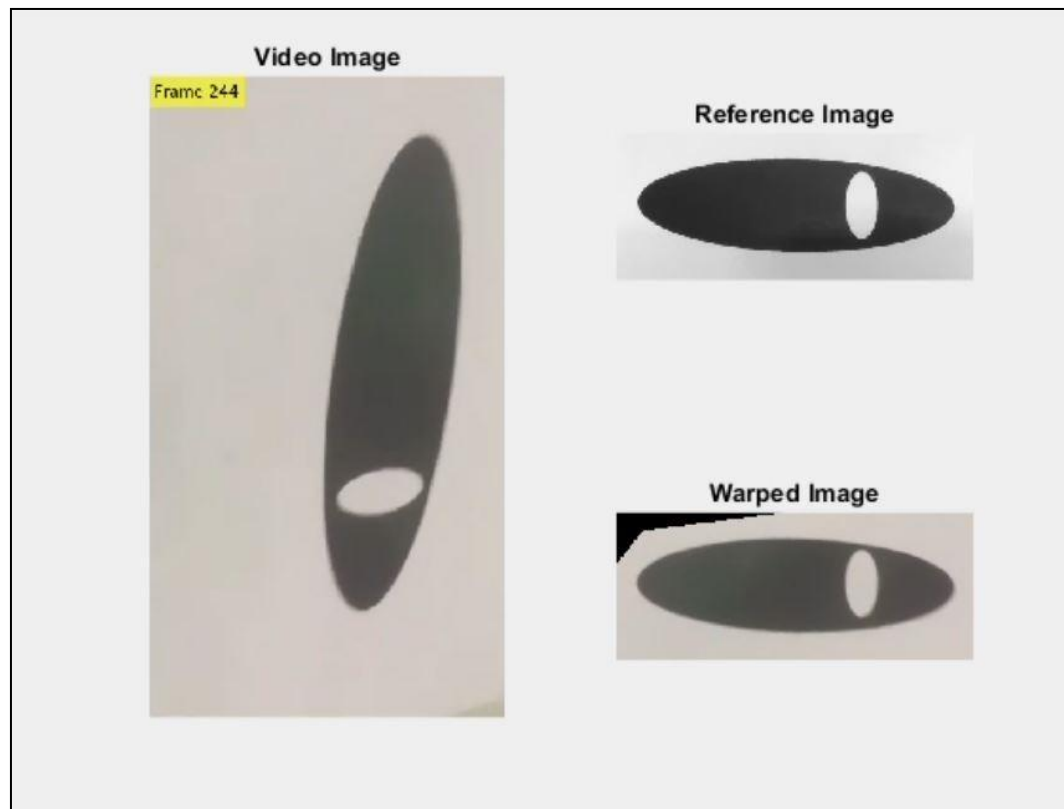


Figure 5.8 - Frame 244 rectification done after using the Observer Estimator

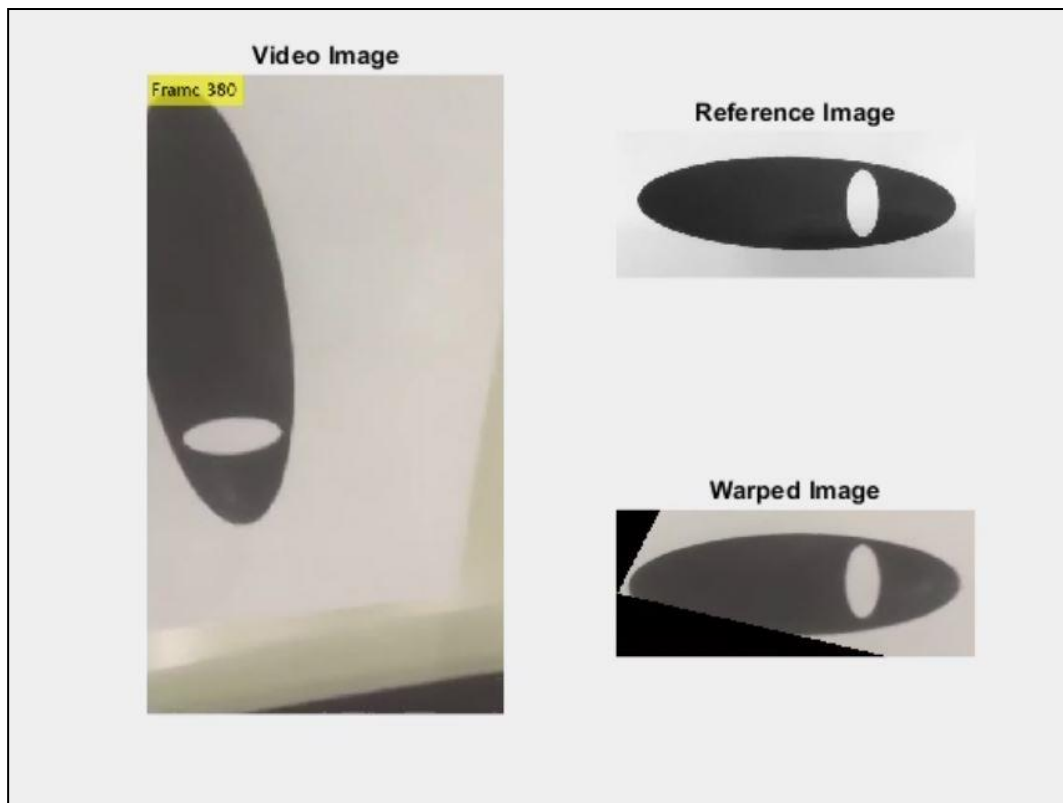


Figure 5.9 - Frame 380 rectification done after using the Observer Estimator.

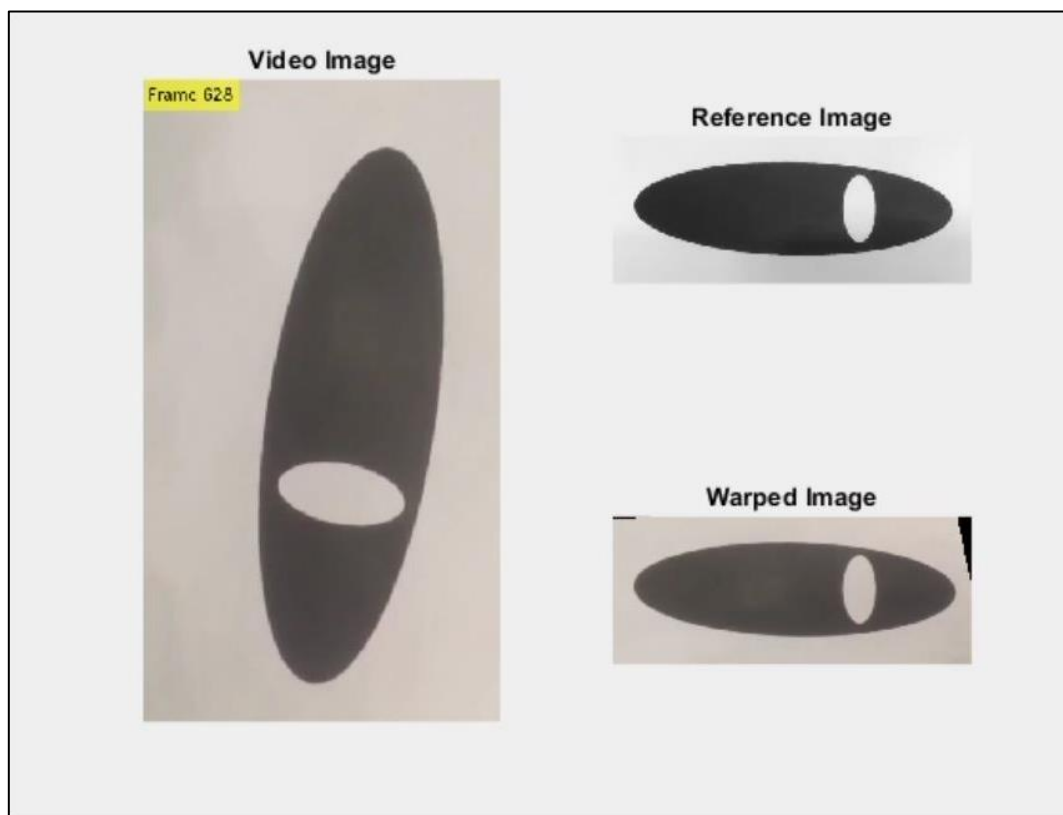


Figure 5.10 - Frame 628 rectification done after using the Observer Estimator.

5.3 Comparison of the two Methods

As can be seen from the images, the implementation of the video rectification is better if done using the observer. Indeed, the presence of an observer improves the estimation of the homography matrix over time. This performances boost is due to the feedback of the previous frame's estimation, that makes computational time and effort lower, compared to the ones obtained by computing algebraically individual raw homography for each image.

However, the Static Estimator produce good results in the same way.

The rectification is possible also in case the two ellipsis are not completely observable situations, like in frame 380.

The full videos of the two implementations could be seen through the following link:

<https://vimeo.com/432409698>

Part II

Modelling of Parrot-Mambo Drone

Chapter 6

Quadrotor Modelling

In the previous chapters different ways of estimating the homography matrix have been designed and simulated using the software “Matlab”. Then, in chapter 5, has been shown how these methods can be implemented in a video. All these steps are preliminary for the final goal of this research, which is the implementation of the homography matrix’s estimation online, using the drone’s operating system. The next step to be done in order to reach this goal is to build a mathematical model that describes the drone’s behaviour using mathematical equations.

In this chapter, the mathematical model of a quadrotor’s dynamics is derived, using Newton’s and Euler’s laws. The output of the model is the position of a point in the raster’s coordinate plane. After the building of the system, the observability is checked in order to start the design of a proper observer for the system of interest.

In the first paragraph, the model and its equations are derived, explaining the theory that is behind them.

In the second paragraph, the observability of the model’s state is studied.

6.1 Quadrotor Mathematical Model

The drone is described by a mathematical model in a state space representation.

The state of the system is the pose of the drone, composed by the position $\rho \in \mathbb{R}^3$ expressed in the inertial frame and the orientation $R \in SO(3)$ described by the rotation matrix from the body frame to the inertial one. The inputs of the model are the linear and the angular velocities, respectively $v \in \mathbb{R}^3$ and $\omega \in \mathbb{R}^3$, of the quadrotor expressed in the drone body frame.

Finally, the outputs of the system are the coordinate of a point expressed in the pixel map of the camera.

In order to write the mathematical model if is necessary to introduce the reference coordinates. In particular, it is possible to use two reference frames, one fixed (called inertial F_I relative to the North-East-Down (NED) frame) and one mobile fixed to the centre of mass of the drone and with (called ABC, Aircraft Body Centre, F_{ABC}). It is needed to find the relationship between the two because the input of the system are given in the body reference frame F_{ABC} and the states are expressed in the inertial reference one F_I .

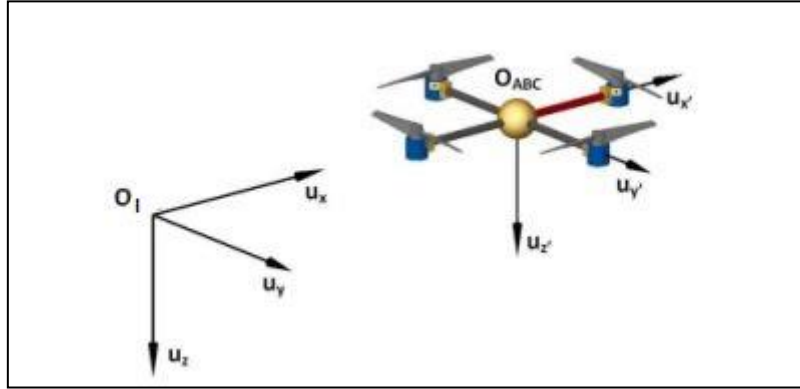


Figure 6.1 – Inertial and mobile reference frames.

6.1.1 Nomenclature

In the following pages this nomenclature is used:

- $\rho = [x, y, z]^T$: position of the drone's centre of mas with respect to the inertial frame;
- $\rho_C = [x_C, y_C, z_C]^T$: position of the camera's eye with respect to the body fixed frame;
- R : rotation matrix from the body fixed frame to the inertial one;
- R_C : rotation matrix from the body fixed frame to the camera one;
- $P = [x_P, y_P, z_P]^T$: position of a generic point with respect to the inertial frame;
- $P_{camera} = [x_{P,camera}, y_{P,camera}, z_{P,camera}]^T$: position of a generic point with respect to the camera frame;
- $P_{normalized} = [x_{P,normalized}, y_{P,normalized}]$: position of a generic point with respect to the NDC frame;
- $P_{raster} = [X, Y]^T$: position of a generic point with respect to the raster frame;
- f : focal length of the camera;

6.1.2 State Equations

In order to represent the attitude of the quadrotor we use rotation matrix R belonging to the 3D rotation group $SO(3)$. In mechanics and geometry, $SO(3)$ is the group of all rotations about the origin of three-dimensional Euclidean space:

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} | R^T R = I, \det(R) = 1\}$$

This choice is preferable than using Euler angles (which have singularities in representation) or quaternions (which might lead to ambiguous control actions due to the phenomenon of unwinding).

The kinematic model of the drone can be described as:

$$\dot{\rho} = Rv$$

$$\dot{R} = RS(w),$$

Where $S(a)$ is the skew-symmetric map defined as:

$$S(a) = -S(a)^T = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

6.1.3 Output Equations

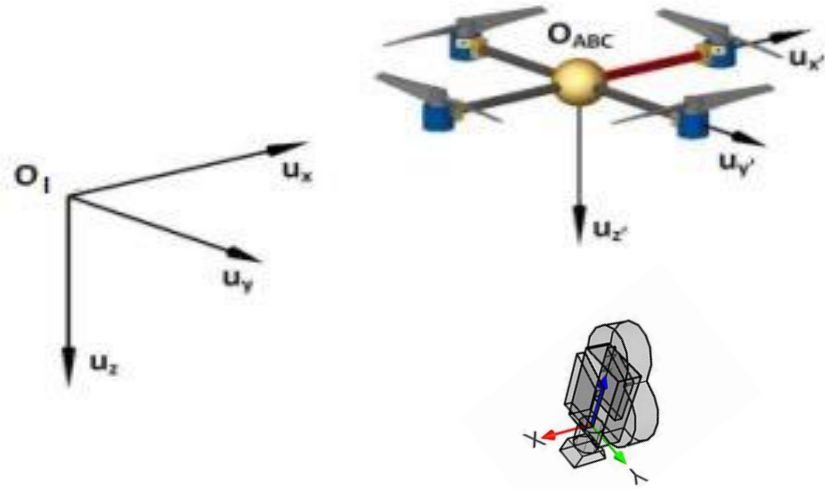


Figure 6.3 – Inertial, mobile and camera reference frames.

The output of the system is the resulting image seen. When a point or vertex is defined in the scene and is visible to the eye or to the camera, it appears as a dot or more precisely as a pixel in the digital image.

Cameras can be studied as any other 3D object, so we have to define a new reference system. The camera coordinate system (also called eye coordinate system) has its origin in the eye of the camera. The z-axis of this frame points to the inside of the camera. Because of this, a point defined in camera space can only be visible, if its z-coordinate is negative.

The camera frame is translated and rotated with respect to the drone's fixed frame. Its pose is expressed by its position $\rho_C = [x_C, y_C, z_C]^T$ and its rotation matrix R_C from the drone's fixed frame to the camera's one.

We want to transform a point's coordinates from world space to camera space. What we want to achieve, is to compute P' , the coordinates of a point P on the surface of a canvas. The canvas (also called the virtual projection plane or the image plane) is the 2D surface on which the image of the scene is drawn. It has a focal distance from the eye of the camera that is considered equal to 1.

If you trace a line from P to the eye, P' is the line's point of intersection with the canvas.

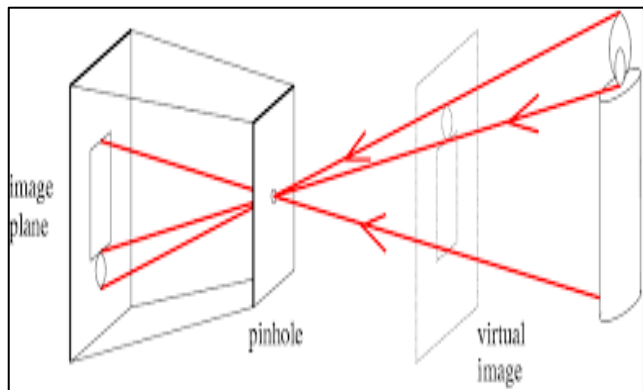


Figure 6.4 – Model of camera image reconstruction.

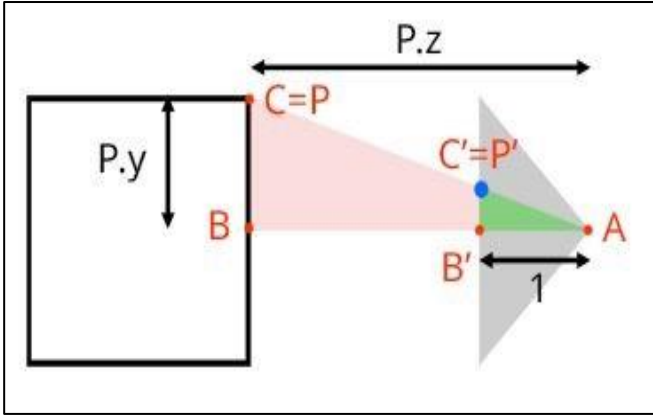


Figure 6.5 – Bird view of world space and camera space.

When the point P coordinates are defined with respect to the camera coordinate system, computing the position of P' is trivial using simple geometrical consideration. The triangles ABC and AB'C' are similar, so:

$$P'(y) = P(y)/P(z)$$

$$P'(x) = P(x)/P(z)$$

This is one of the most fundamental relation in computer graphics, known as the perspective divide. If the focal length was different from one and equal to f we would have:

$$P'(y) = (P(y) * f)/P(z)$$

$$P'(x) = (P(x) * f)/P(z)$$

We can now compute the camera-to-world matrix simply by the product of the translation matrices and the rotation ones. This matrix can be found using the states and the known camera's parameter:

- $T_R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$: rotation from the inertial frame to the drone fixed frame;
- $T_\rho = \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix}$: translation from the inertial frame to the drone fixed frame;
- $T_{RC} = \begin{bmatrix} r_{C,11} & r_{C,12} & r_{C,13} & 0 \\ r_{C,21} & r_{C,22} & r_{C,23} & 0 \\ r_{C,31} & r_{C,32} & r_{C,33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$: rotation from the drone fixed frame to the camera frame;
- $T_{\rho C} = \begin{bmatrix} 1 & 0 & 0 & -x_C \\ 0 & 1 & 0 & -y_C \\ 0 & 0 & 1 & -z_C \\ 0 & 0 & 0 & 1 \end{bmatrix}$: translation from drone fixed frame to the camera frame;
- $T_{tot} = T_{RC}T_{\rho C}T_RT_\rho$: transformation matrix from inertial frame to camera frame;

We can now transform any point of which the coordinates are defined in world space to camera space, by multiplying this point by the world-to-camera matrix:

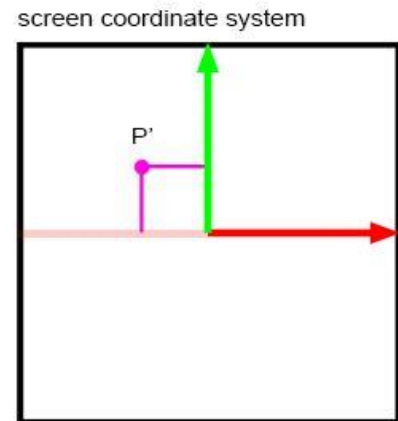
$$P_{camera} = P * T_{tot}$$

And so:

$$P'(y) = P_{camera}(y)/P_{camera}(z)$$

$$P'(x) = P_{camera}(x)/P_{camera}(z)$$

Now we have coordinates expressed in the screen coordinate system.



Since P' is a 2D point, it is defined with respect to a 2D coordinate system called the image or screen coordinate system. Images are not infinite in size; they have a width and a height. Thus, a rectangular shape is cut off centred around the image coordinate system defined as the "bounded region" over which the image of the 3D scene is drawn. Any projected point whose absolute x and y coordinate is greater than half of the canvas' width or half of the canvas' height respectively is not visible in the image.

If P' is visible it should appear as a dot in the image. A dot in a digital image is a pixel. Pixels are 2D points, and their coordinates are integers. This coordinate system is called raster coordinate system and is located in the upper-left corner of the image. Its x-axis points to the right, and its y-axis points downwards. A pixel in this coordinate system, is one unit long in x and y. What we need to do, is convert P' coordinates which are defined with respect to the image or screen coordinate system into pixel coordinates.

The first thing we are going to do is to remap P' coordinates in the range $[0,1]$, normalizing the coordinate:

$$P_{normalized}(x) = (P'(x) + width/2)/width$$

$$P_{normalized}(y) = (P'(y) + height/2)/height$$

The coordinate system in which the points are defined after normalization is the NDC (Normalized Device Coordinate) coordinate system or NDC space. The NDC coordinate system's origin is situated in the lower-left corner of the canvas.

The last step is multiply the projected point x- and y-coordinates in NDC space, by the actual image pixel width and pixel height respectively. This is a simple remapping of the range $[0,1]$ to the range $[0, Pixel_Width]$ for the x-coordinate, and $[0, Pixel_Height]$ for the y-coordinate respectively. Pixel coordinates are integers, so rounding off the resulting numbers to the smallest following integer value is necessary. This is done using the floor function $\lfloor - \rfloor$:

$$P_{raster}(x) = \lfloor P_{normalized}(x) * Pixel_Width \rfloor$$

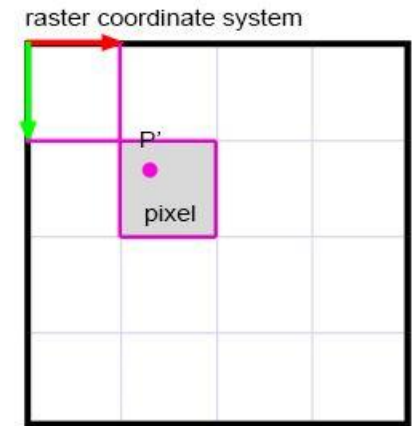
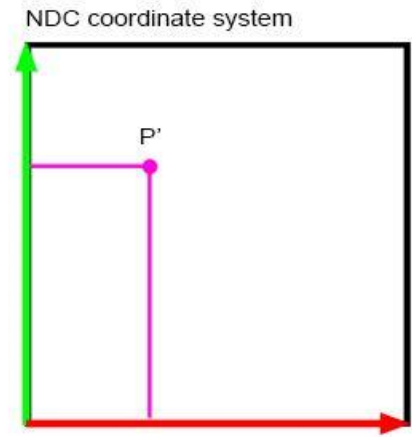
$$P_{raster}(y) = \lfloor P_{normalized}(y) * Pixel_Height \rfloor$$

There is a final small detail though we need to take care of. The y-axis in the NDC coordinate system points up while in the raster coordinate system points down. Thus, to go from one coordinate system to the other, P' y-coordinate also needs to be inverted. We can easily account for this by doing a small modification to the above equations:

$$P_{raster}(x) = X = \lfloor P_{normalized}(x) * Pixel_Width \rfloor$$

$$P_{raster}(y) = Y = \lfloor (1 - P_{normalized}(y)) * Pixel_Height \rfloor$$

After this final step, coordinates are defined in the raster space.



6.1.4 Total System

Variables:

$$\rho = [x \quad y \quad z]'; \quad (\text{In the inertial reference frame})$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix};$$

$$R^{-1} = R^T = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix}; \quad (\text{From the definition of SO(3)})$$

$$T_{tot} = T_{RC} T_{\rho C} T_R T_\rho = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} =$$

$$\begin{bmatrix} (r_{11}r_{C,11} + r_{12}r_{C,12} + r_{13}r_{C,13}) & (r_{21}r_{C,11} + r_{22}r_{C,12} + r_{23}r_{C,13}) & (r_{31}r_{C,11} + r_{32}r_{C,12} + r_{33}r_{C,13}) & x_C - t_{11}x - t_{12}y - t_{13}z \\ (r_{11}r_{C,21} + r_{12}r_{C,22} + r_{13}r_{C,23}) & (r_{21}r_{C,21} + r_{22}r_{C,22} + r_{23}r_{C,23}) & (r_{31}r_{C,21} + r_{32}r_{C,22} + r_{33}r_{C,23}) & y_C - t_{21}x - t_{22}y - t_{23}z \\ (r_{11}r_{C,31} + r_{12}r_{C,32} + r_{13}r_{C,33}) & (r_{21}r_{C,31} + r_{22}r_{C,32} + r_{23}r_{C,33}) & (r_{31}r_{C,31} + r_{32}r_{C,32} + r_{33}r_{C,33}) & z_C - t_{31}x - t_{32}y - t_{33}z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

States & Output:

$$x_1 = x; \quad x_2 = y; \quad x_3 = z; \quad (==\rho)$$

$$x_4 = r_{11}; \quad x_5 = r_{12}; \quad x_6 = r_{13};$$

$$x_7 = r_{21}; \quad x_8 = r_{22}; \quad x_9 = r_{23}; \quad (==R)$$

$$x_{10} = r_{31}; \quad x_{11} = r_{32}; \quad x_{12} = r_{33};$$

$$u_1 = v_x; \quad u_2 = v_y; \quad u_3 = v_z; \quad (==v)$$

$$u_4 = \omega_x; \quad u_5 = \omega_y; \quad u_6 = \omega_z \quad (==\omega)$$

$$y_1 = X; \quad y_2 = Y;$$

State equations:

$$\begin{aligned}
 \dot{x}_1 &= x_4 u_1 + x_5 u_2 + x_6 u_3; \\
 \dot{x}_2 &= x_7 u_1 + x_8 u_2 + x_9 u_3; \\
 \dot{x}_3 &= x_{10} u_1 + x_{11} u_2 + x_{12} u_3; \\
 \dot{x}_4 &= x_5 u_6 - x_6 u_5; & \dot{x}_5 &= -x_4 u_6 + x_6 u_4; & \dot{x}_6 &= x_4 u_5 - x_5 u_4; \\
 \dot{x}_7 &= x_8 u_6 - x_9 u_5; & \dot{x}_8 &= -x_7 u_6 + x_9 u_4; & \dot{x}_9 &= x_7 u_5 - x_8 u_4; \\
 \dot{x}_{10} &= x_{11} u_6 - x_{12} u_5; & \dot{x}_{11} &= -x_{10} u_6 + x_{12} u_4; & \dot{x}_{12} &= x_{10} u_5 - x_{11} u_4;
 \end{aligned}
 \tag{=\dot{\rho}}$$

Output Equation:

$$\begin{aligned}
 y_1 = X &= \lfloor P_{normalized}(x) * Pixel_{width} \rfloor = \left\lfloor \frac{(P'(x) + width/2)}{width} * Pixel_{width} \right\rfloor = \\
 &= \left\lfloor \frac{(\frac{P_{camera}(x)}{-P_{camera}(z)} + width/2)}{width} * Pixel_{width} \right\rfloor = \\
 &= \left\lfloor \frac{\frac{t_{11}*x_P + t_{12}*y_P + t_{13}*z_P + t_{14}}{-(t_{31}*x_P + t_{32}*y_P + t_{33}*z_P + t_{34})} + \frac{width}{2}}{width} * Pixel_{width} \right\rfloor \\
 y_2 = Y &= \lfloor (1 - P_{normalized}(y)) * Pixel_{Height} \rfloor = \left\lfloor \left(1 - \frac{(P'(y) + Height/2)}{Height}\right) * Pixel_{Height} \right\rfloor = \\
 &= \left\lfloor \left(1 - \frac{(\frac{P_{camera}(y)}{-P_{camera}(z)} + Height/2)}{Height}\right) * Pixel_{Height} \right\rfloor = \\
 &= \left\lfloor \left(1 - \frac{\frac{t_{21}*x_P + t_{22}*y_P + t_{23}*z_P + t_{24}}{-(t_{31}*x_P + t_{32}*y_P + t_{33}*z_P + t_{34})} + \frac{Height}{2}}{Height}\right) * Pixel_{Height} \right\rfloor;
 \end{aligned}$$

After some simplifications, the system could be reduced in the form:

$$y_1 = \frac{x_4 * k_1 + x_7 * k_2 + x_{10} * k_3 - x_1}{x_6 * k_1 + x_9 * k_2 + x_{12} * k_3 - x_3} * C_1 + C_2 ;$$

$$y_2 = \frac{x_5 * k_1 + x_8 * k_2 + x_{11} * k_3 - x_2}{x_6 * k_1 + x_9 * k_2 + x_{12} * k_3 - x_3} * C_3 + C_4 ;$$

where $k_{1,2,3}$ and $C_{1,2,3,4}$ are known constants. ($y_{1,2}$ should be rounded to the closer next integer to represent the pixel coordinate)

6.2 Observability Analysis

The system studied has 12 states, 3 coordinates of the body frame origin expressed in the inertial reference frame and 9 elements of the rotation matrix. However the rotation matrix R belongs to the special orthogonal group in three dimension $SO(3)$, and so:

$$R^T R = I$$

$$\begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

From this definition we can obtain the following equations:

$$\langle u_i, u_j \rangle = 0, \quad i = 1,2,3; j = 1,2,3; \forall i \neq j$$

$$\langle u_i, u_j \rangle = 0, \quad i = 1,2,3; j = 1,2,3; \forall i = j$$

Where $u_i, i = 1,2,3$ are the rows of R . These constraints build a system of 6 equations making the degrees of freedom only 3. It is possible to add another equation to the set because $\det(R) = 1$, but it wouldn't add any information about the elements.

Now we know the model has 6 independent variables. In order to study the observability we could compute the Lie derivative of the output with respect to the dynamic of the system. Calling $h(x, u) = y_1(x, u)$ and $f(x, u) = \dot{x}(x, u)$, and considering $u=0$:

$$L_f^1 h = 0;$$

This is predictable because if the drone is still ($u=0$) the coordinate of a point in the raster plane would not change. The same reasoning should be done for the Y coordinate y_2 . This means that if we have as output of the system only the two coordinate in the raster plane of a single point the system is unobservable.

However, in order to make the system observable is possible to add more outputs, in particular, we need 4 more equations. This could be done have two more points. The output equations are:

$$y_1 = \frac{x_4 * k_1 + x_7 * k_2 + x_{10} * k_3 - x_1}{x_6 * k_1 + x_9 * k_2 + x_{12} * k_3 - x_3} * C_1 + C_2 ;$$

$$y_2 = \frac{x_5 * k_1 + x_8 * k_2 + x_{11} * k_3 - x_2}{x_6 * k_1 + x_9 * k_2 + x_{12} * k_3 - x_3} * C_3 + C_4 ;$$

$$y_3 = \frac{x_4 * k_4 + x_7 * k_5 + x_{10} * k_6 - x_1}{x_6 * k_4 + x_9 * k_5 + x_{12} * k_6 - x_3} * C_5 + C_6 ;$$

$$y_4 = \frac{x_5 * k_4 + x_8 * k_5 + x_{11} * k_6 - x_2}{x_6 * k_4 + x_9 * k_5 + x_{12} * k_6 - x_3} * C_7 + C_8 ;$$

$$y_5 = \frac{x_4 * k_7 + x_7 * k_2 + x_{10} * k_3 - x_1}{x_6 * k_7 + x_9 * k_2 + x_{12} * k_3 - x_3} * C_9 + C_{10} ;$$

$$y_6 = \frac{x_5 * k_7 + x_8 * k_8 + x_{11} * k_9 - x_2}{x_6 * k_7 + x_9 * k_8 + x_{12} * k_9 - x_3} * C_{11} + C_{12} ;$$

If these equations are linear independent the system became observable a feedback observer could be designed. This could be d

Chapter 7

Conclusions

This chapter summarizes the main researches done and contain the final conclusion from this thesis work in Section 7.1. In Section 7.2 the resulting topics for future research are discussed.

7.1 Conclusive Considerations

Autonomous, or self-driving, vehicles are becoming more and more common in our society. Such a new and emerging technology that has the potential of completely reinventing the way we live. A subgroup of the autonomous vehicle is the unmanned aerial vehicle (UAV), commonly known as a drone. My thesis work is a part of a bigger and more elaborated project which final goal is the complete automation of a Parrot Mambo drone's flight.

A crucial point for the automation of a vehicle is the positioning process. The purpose of my research is to achieve an online Local positioning on a Mambo Parrot fly drone. In particular, the on-board camera is the chosen sensor used in order to achieve the final goal and the homography is the selected technique.

Homography is a technique widely used in computer vision, that takes the name from the homonymous matrix that expresses the geometrical transformation between two images of the same view, with different points of view. The main idea is that knowing the homography between a Reference View and the instantaneous camera image shown captured by the drone's camera, it would be possible to compute the pose of the quadroto, and so it's position and attitude with respect to an inertial and known reference frame.

The first instinctive step to perform in order to achieve this goal is the estimation of the homography matrix that links two views of the same scene. In order to do this, a set of equation able to solve the eight degrees of freedom of the homography matrix. This could be done with a proper choice of the Reference Image. The first choice was the use of rectangular shapes that highlight their corners. These points correspondences build a set of linearly independent equations able to solve the homography matrix. An estimator is designed and tested, revealing good results. Nevertheless, this approach is difficult to apply in real situation, because of the not negligible presence of rectangular shapes that could trick our operating system.

The solution adopted in this project is the change of the Reference Image, making its detection easier and more reliable. The second and final Reference Image chosen represent two conics, in particular two ellipsis. Using this new Reference Image the estimation is not done anymore using points

correspondences but conics' ones. The new approach needed more effort for the computation of the homography matrix parameters, but this problem is solved by the usage of a first-order Taylor expansion. As before, an estimator is designed and tested, showing good results in simple situation, but, especially when the distortion of the shapes due to the perspective is high, the estimation accuracy decrease. This new problem is faced changing the estimation process, from a static one to a time evolving one. An observer estimator is designed.

Until now the estimation developed consider the homography as an incidental variable and are not focused on improving the estimation over time. The quality of the homography estimates depends heavily on the nature of the image features exploited as well as the algorithm used without increasing the quality of the estimation over time. The methodology taken exploits the underlying structure of the Special Linear group $SL(3)$, a Lie group isomorphic to the group of homographies, designing a State Observer in discrete time. The observer is designed and tested, performing the best results obtained compare with the ones of the previous estimator.

All the previous researches have been implemented for estimating the homography matrix linking two pictures, for a static case, but the drone's camera is not static and is actually filming a Video. A new approach in order to perform what is called image stabilization using the homography technique should be designed. Initially the algorithm was tested using the Static estimator for each single frame. This would be an effort too heavy for the drone's operating system, and the rectified video produced is not smooth as expected. The best solution is to merge the Static and the Observer Estimator, exploiting the best from each. The Static Estimator was used in order to estimate the homography for the first frame, then the Observer is used to perform the estimation for the following ones, increase the quality of the estimation over time. This implementation has produced very good and satisfactory results.

All these researches are preliminary for the final goal, because they can not be applied directly to the drone's operating system. The next step to be done is to build a mathematical model that describes the drone's behaviour using mathematical equations. In Chapter 6, the mathematical model of a quadrotor's dynamics is derived, using Newton's and Euler's laws. The output of the model is the position of a point in the raster's coordinate plane. After the modelling of the system, the observability is checked.

7.2 Recommendations

This thesis develop the homography matrix estimation using points' and conics' correspondences. The different estimator has been designed and tested in static and non-static case. These experimental tests have been taken using pre-recorded and pre-taken, providing satisfying results. However, this options would not take into account the problem of having an online estimation on the drone's operating system. Moreover, the code developed should be translated in the language of the operating system. Therefore, a possible way of continuing this work could be use the code generation process to generate a proper code downloadable on the drone, and use experimental tests of it.

Moreover, this thesis research are limited in the homography estimation. An important step to be developed should be the decomposition of the homography matrix, knowing the camera's specification. This would allow to find the position and the attitude of the Moved Camera, with respect to the Reference one.

Furthermore, it was discussed a new modelling procedure in order to describe how the pose of the drone changes the position of a point in the raster coordinate system. The originality of this modelling relays in the choice of the attitude representation and in the selected outputs. The observability of this model has been checked, finding that is possible to have an estimation of the states (attitude and position) adding more outputs (in particular using three points). A possible future approach should be study the modelling of this system with more accuracy, finding a way to estimate the observation of the states limiting the points' used.

Finally, the scope of this thesis was limited on the homography technique in order to achieve the Local Positioning of an autonomous quadrotor. Another approach could be the implementation of a state observer designed using the mathematical model cited before.

*By far the greatest danger of Artificial Intelligence
is that people conclude too early that they understand it
(Eliezer Yudkowsky)*

Appendix A

Proofs and Methods

A.1 Single Value Decomposition (SVD)

In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix that generalizes the eigendecomposition of a square normal matrix to any $m \times n$ matrix via an extension of the polar decomposition.

Specifically, the singular value decomposition of an $m \times n$ real or complex matrix M is a factorization of the form $U\Sigma V^*$, where U is an $m \times m$ real or complex unitary matrix, Σ is an $m \times n$ matrix with non-negative real numbers on the diagonal, and V is an $n \times n$ real or complex unitary matrix. If M is real, U and $V^T = V^*$ are real orthogonal matrices.

The diagonal entries of Σ are known as the singular values of M . The number of non-zero singular values is equal to the rank of M . The columns of U and the columns of V are called the left-singular vectors and right-singular vectors of M , respectively.

A.2 “Harris Corner Detection” Algorithm

Harris Corner Detector is a corner detection operator that is commonly used in computer vision algorithms to extract corners and infer features of an image. It was first introduced by Chris Harris and Mike Stephens in 1988 upon the improvement of Moravec's corner detector. Compared to the previous one, Harris' corner detector takes the differential of the corner score into account with reference to direction directly, instead of using shifting patches for every 45 degree angles, and has been proved to be more accurate in distinguishing between edges and corners. Since then, it has been improved and adopted in many algorithms to preprocess images for subsequent applications.

Without loss of generality, is assumed a grayscale 2-dimensional image is used. Let this image be given by I . Consider taking an image patch $(x, y) \in W(\text{window})$ and shifting it $(\Delta x, \Delta y)$. The sum of squared differences (SSD) between these two patches, denoted f , is given by:

$$f(\Delta x, \Delta y) = \sum_{(x_k, y_k)} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$

$I(x_k + \Delta x, y_k + \Delta y)$ can be approximated by a Taylor expansion. Let I_x and I_y be the partial derivatives of I , such that

$$I(x + \Delta x, y + \Delta y) \approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y$$

This produces the approximation

$$f(\Delta x, \Delta y) \approx \sum_{(x_k, y_k)} (I_x(x, y)\Delta x + I_y(x, y)\Delta y)^2$$

which can be written in matrix form:

$$f(\Delta x, \Delta y) \approx (\Delta x, \Delta y) M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

where M is the structure tensor,

$$M = \sum_{(x_k, y_k)} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_{(x_k, y_k)} I_x^2 & \sum_{(x_k, y_k)} I_x I_y \\ \sum_{(x_k, y_k)} I_x I_y & \sum_{(x_k, y_k)} I_y^2 \end{bmatrix}.$$

A.3 Least Square Method

The method of least squares is a standard approach in regression analysis to approximate the solution of overdetermined systems (sets of equations in which there are more equations than unknowns) by minimizing the sum of the squares of the residuals made in the results of every single equation.

In this thesis is used the ordinary least squares (OLS) is a type of linear least squares method for estimating the unknown parameters in a linear regression model.

Consider an overdetermined system

$$\sum_{j=1}^p X_{ij}\beta_j = y_i, (i = 1, 2, \dots, n),$$

of n linear equations in p unknown coefficients, $\beta_1, \beta_2, \dots, \beta_p$, with $n > p$. This can be written in matrix form as

$$\mathbf{X}\boldsymbol{\beta} = \mathbf{y},$$

where

$$\mathbf{X} = \begin{bmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{np} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix}.$$

Such a system usually has no exact solution, so the goal is instead to find the coefficients $\boldsymbol{\beta}$ which fit the equations "best", in the sense of solving the quadratic minimization problem

$$\hat{\boldsymbol{\beta}} = \arg \min S(\boldsymbol{\beta})$$

where the objective function S is given by

$$S(\boldsymbol{\beta}) = \sum_{i=1}^n \left| y_i - \sum_{j=1}^p X_{ij}\beta_j \right|^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

A justification for choosing this criterion is given in Properties below. This minimization problem has a unique solution, provided that the p columns of the matrix \mathbf{X} are linearly independent, given by solving the normal equations

$$(\mathbf{X}^T \mathbf{X})\hat{\boldsymbol{\beta}} = \mathbf{X}^T \mathbf{y}.$$

The matrix $\mathbf{X}^T \mathbf{y}$ is known as the moment matrix of regressand by regressors.[1] Finally,

$\hat{\boldsymbol{\beta}}$ is the coefficient vector of the least-squares hyperplane, expressed as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

Armiko back-tracking procedure

Appendix B

Glossary

M_{in} : transformation matrix with intrinsic parameters;

M_{ex} : transformation matrix with extrinsic parameters;

x : position wrt the x axis;

y : position wrt the y axis;

z : position wrt the z axis;

e : eccentricity of a conic;

C_Q : matrix of the quadratic;

ρ : position of the drone's centre of mass with respect to the inertial frame;

ρ_C : position of the camera's eye with respect to the body fixed frame;

R : rotation matrix from the body fixed frame to the inertial one;

R_C : rotation matrix from the body fixed frame to the camera one;

P : position of a generic point with respect to the inertial frame;

P_{camera} : position of a generic point with respect to the camera frame;

$P_{normalized}$: position of a generic point with respect to the NDC frame;

P_{raster} : position of a generic point with respect to the raster frame;

f : focal length of the camera;

Appendix C

Code Lines

```
%% INITIALIZATION

global K;
clear all, close all, clc;

alpha=0.005; beta=0.75; sigma=0.25; K=diag([1,1,2]);

%% REFERENCE IMAGE FEATURES EXTRACTION

Img1=rgb2gray(imread('Reference.jpg'));
BWperimref=bwperim(Img1<150);
[B,~,~,A] = bwboundaries(BWperimref,'noholes');

for i=1:length(B)

    enclosing_boundary = find(A(i,:));
    enclosed_boundaries = find(A(:,i));
    if length(B{i}(:,1))>50 && length(enclosing_boundary)==1 &&
isempty(enclosed_boundaries) && length(find(A(:,enclosing_boundary)))==1

        e1=enclosing_boundary;
        e2=i;

    end
end

[a1,b1,phi1,Xc1,Yc1] = fit_ellipse_light( B{e1}(:,2),B{e1}(:,1) )
[a2,b2,phi2,Xc2,Yc2] = fit_ellipse_light( B{e2}(:,2),B{e2}(:,1) )

[C1,phi1] = fit_ellipse_light1( (B{e1}(:,2)-Xc1)./300, (B{e1}(:,1)-Yc1)./300 );
[C2,phi2] = fit_ellipse_light1( (B{e2}(:,2)-Xc1)./300, (B{e2}(:,1)-Yc1)./300 );

C1=C1./(abs(det(C1))^(1/3));
C2=C2./(abs(det(C2))^(1/3));

C(:, :, 3)=C1;
C(:, :, 4)=C2;

%%

obj=VideoReader('Video.mp4');
vid=VideoWriter('NewVideoObserver.mp4','MPEG-4');%use h264 encoding
open(vid);
```

```

Imov = read(obj,1);
Img2=rgb2gray(Imov);
BWperimov=bwperim(Img2<150);
[Bmov,~,~,Amov] = bwboundaries(BWperimov,'noholes');

for i=1:length(Bmov)

    enclosing_boundary = find(Amov(i,:));
    enclosed_boundaries = find(Amov(:,i));
    if length(Bmov{i}(:,1))>50 && length(enclosing_boundary)==1 &&
isempty(enclosed_boundaries) && length(find(Amov(:,enclosing_boundary)))==1

        emov1=enclosing_boundary;
        emov2=i;

    end
end

[amov1,bmov1,phimov1,Xcmov1,Ycmov1] = fit_ellipse_light(
Bmov{emov1}(:,2),Bmov{emov1}(:,1) )
[amov2,bmov2,phimov2,Xcmov2,Ycmov2] = fit_ellipse_light(
Bmov{emov2}(:,2),Bmov{emov2}(:,1) )

[Cmov1,phimov1] = fit_ellipse_light1( (Bmov{emov1}(:,2)-
Xcmov1)./300, (Bmov{emov1}(:,1)-Ycmov1)./300 );
[Cmov2,phimov2] = fit_ellipse_light1( (Bmov{emov2}(:,2)-
Xcmov1)./300, (Bmov{emov2}(:,1)-Ycmov1)./300 );
Cmov1=Cmov1./(abs(det(Cmov1))^(1/3));
Cmov2=Cmov2./(abs(det(Cmov2))^(1/3));

T1=[1,0,0; 0, 1, 0; -Xc1, -Yc1, 1]';
T2=[1,0,0; 0, 1, 0; -Xcmov1, -Ycmov1, 1]';

S1=[300, 0, 0; 0, 300, 0; 0, 0, 1];
S2=[300, 0, 0; 0, 300, 0; 0, 0, 1];

H1 = hg_2elin(Cmov1,Cmov2,C1,C2);
H1=-H1./(abs(det(H1))^(1/3));
H=inv(T1)*S2*H1*inv(S1)*T2;
H=H./(abs(det(H))^(1/3));
H=H./H(3,3);

H_hat(:, :, 1)=H1;
%%

for img=1:obj.NumberOfFrames;

    img
    frames{img}=read(obj,img);
    Imov = frames{img};
    Img2=rgb2gray(Imov);
    BWperimov=bwperim(Img2<150);
    [Bmov,~,~,Amov] = bwboundaries(BWperimov,'noholes');

    for i=1:length(Bmov)

        enclosing_boundary = find(Amov(i,:));
        enclosed_boundaries = find(Amov(:,i));
        if length(Bmov{i}(:,1))>50 && length(enclosing_boundary)==1 &&
isempty(enclosed_boundaries) && length(find(Amov(:,enclosing_boundary)))==1

            emov1=enclosing_boundary;

```

```

        emov2=i;

    end
end

[amov1,bmov1,phimov1,Xcmov1,Ycmov1] = fit_ellipse_light(
Bmov{emov1}(:,2),Bmov{emov1}(:,1) )
[amov2,bmov2,phimov2,Xcmov2,Ycmov2] = fit_ellipse_light(
Bmov{emov2}(:,2),Bmov{emov2}(:,1) )

[Cmov1,phimov1] = fit_ellipse_light1( (Bmov{emov1}(:,2)-
Xcmov1)./300,(Bmov{emov1}(:,1)-Ycmov1)./300 );
[Cmov2,phimov2] = fit_ellipse_light1( (Bmov{emov2}(:,2)-
Xcmov1)./300,(Bmov{emov2}(:,1)-Ycmov1)./300 );

Cmov1=Cmov1./(abs(det(Cmov1))^(1/3));
Cmov2=Cmov2./(abs(det(Cmov2))^(1/3));

T2=[1,0,0; 0, 1, 0; -Xcmov1, -Ycmov1, 1]';

%% OBSERVER

C(:, :, 1)=Cmov1;
C(:, :, 2)=Cmov2;

for k=1:1000

    k;

    ek1(:, :, k)=inv(H_hat(:, :, k))'*C(:, :, 1)*inv(H_hat(:, :, k)); %->M1=C(:, :, 3)
    ek2(:, :, k)=inv(H_hat(:, :, k))'*C(:, :, 2)*inv(H_hat(:, :, k)); %->M2=C(:, :, 4)

    innovation(:, :, k)=-projection_operator(...
        ek1(:, :, k)*(ek1(:, :, k)-C(:, :, 3))*K+ek1(:, :, k)*K*(ek1(:, :, k)-
C(:, :, 3))+...
        ek2(:, :, k)*(ek2(:, :, k)-C(:, :, 4))*K+ek2(:, :, k)*K*(ek2(:, :, k)-
C(:, :, 4))); %->0

    clear tempm;
    stat=1;
    i=2; tempm(1)=1; tempm(i)=2; min(1)=1; min(2)=1;

    while stat~=0 && tempm(i)~=Inf

        t(k)=alpha*((beta)^tempm(i));
        H_hat(:, :, k+1)=(expm(-innovation(:, :, k).*t(k)))*H_hat(:, :, k);
%->H

        AGG(k)=aggcost(H_hat(:, :, k),C,K);
        AGG(k+1)=aggcost(H_hat(:, :, k+1),C,K);
        diff(k)=AGG(k)-AGG(k+1);
        kost(k)=sigma*alpha*trace(innovation(:, :, k)*innovation(:, :, k)');
        cond(k)=diff(k)/kost(k)-t(k);

        if cond(k)>=0
            stat =-1;
        end

        if stat == 1

            tempm(i+1)=2*tempm(i);

```

```

        min(i+1)=tempm(i);

    end

    if stat == -1

        if cond(k)>0

            tempm(i+1)=tempm(i)-((tempm(i)-min(i))/2);
            min(i+1)=min(i);

        else

            tempm(i+1)=tempm(i)+(abs(tempm(i-1)-tempm(i))/2);
            min(i+1)=tempm(i);

        end

        if abs(tempm(i)-tempm(i-1))==1

            m(k)=min(i+1)+1;
            stat=0;

        end

    end

    i=i+1;

end

t(k)=alpha*((beta)^m(k));
H_hat(:, :, k+1)=expm(-innovation(:, :, k).*t(k))*H_hat(:, :, k);           %->H
% E(:, :, k+1)=H_hat(:, :, k+1)*inv(H);
end

Htot=inv(T1)*S2*H_hat(:, :, end)*inv(S1)*T2;
Htot=Htot./(abs(det(Htot))^(1/3));
Htot=Htot./Htot(3,3);

dim=imref2d(size(Img1));
Htform=projective2d(transpose(Htot));
[out, ref] = imwarp(Imov,Htform,'OutputView',dim);

figure(1), hold on;
Ref=insertText(frames{img},[0,0],['Frame ', num2str(img)], 'FontSize',15);
subplot(1,2,1), imshow(Ref); title('Video Image');
subplot(2,2,2), imshow(Img1); title('Reference Image');
subplot(2,2,4), imshow(out); title('Warped Image');

H_hat(:, :, 1)=H_hat(:, :, end);

frame=getframe(gcf);
writeVideo(vid,frame);

end

close(vid);

```

References

- E. Malis & M. Vargas, *"Deeper understanding of the homography decomposition for vision-based control"*;
- F. Sabatino, *"Quadrotor control: modeling, nonlinear control design, and simulation"*;
- A.A.J. Lefeber, *"Tracking Control of Nonlinear Mechanical Systems"*;
- M.D. Hua, T. Hamel, R. Mahony, G. Allibert, *"Explicit Complementary Observer Design on Special Linear Group $SL(3)$ for Homography Estimation using Conic Correspondences"*;
- E. Lefeber, S.J.A.M. van den Eijnden and H. Nijmeijer, *"Almost Global Tracking Control of a Quadrotor UAV on $SE(3)$ "*;
- E. Lefeber, M.F.A. van deWesterlo, H.Nijmeijer, *"Almost global decentralised formation tracking for multiple distinct UAVs"*;
- H. Wu & Q. Chen, *"Homography from Conic Intersection: Camera Calibration based on Arbitrary Circular Patterns"*;
- R. Marino and P. Tomei, *"Nonlinear Control Design"*;
- J. Kannala, M. Salo and J. Heikkila, *"Algorithms for Computing a Planar Homography from Conics in Correspondence"*;
- C. Novara, *"Nonlinear Control and Aerospace Applications: lecture notes"*, Politecnico di Torino, 2019;
- H.K. Khalil, *"Nonlinear Systems"*, Prentice Hall, 2002;
- M. Violante, M. Velardocchia, M. Canale, *"Technologies for Autonomous Veichles: lecture notes"*, Politecnico di Torino, 2019;
- Tarek Hamel, Robert Mahony, Jochen Trumpf, Pascal Morin and Minh-Duc Hua, *"Homography estimation on the Special Linear Group based on direct point correspondence"*
- O. Chum & J. Matas, *"Homography Estimation from Correspondences of Local Elliptical Features"*;
- O. Chum, T. Pajdla, and P. Sturm, *"The geometric error for homographies"*;
- E. Lefeber, M. Greiff, A. Robertsson, *"Filtered Output Feedback Tracking Control of a Quadrotor UAV"*
- A. Sugimoto, *"A Linear Algorithm for Computing the Homography from Conics in Correspondence"*;
- B. Bona, *"Dynamic Modelling of Mechatronic System"*;
- https://en.wikipedia.org/wiki/Conic_section;
- <https://en.wikipedia.org/wiki/Ellipse>;

<https://en.wikipedia.org/wiki/Homography>;

https://en.wikipedia.org/wiki/Singular_value_decomposition;

<https://www.scratchapixel.com/lessons/3d-basic-rendering/computing-pixel-coordinates-of-3d-point/mathematics-computing-2d-coordinates-of-3d-points>;

Epilogue

Non è facile, quando qualcosa finisce, dire addio. Congedare in poche, umide parole una parte della vita che con la sua mano ti sta salutando, fermamente convinta ad abbandonarti e a non tornare mai più. Così cerchi di fare ordine nella tua mente, ma, proprio in quel momento, tutti quei ricordi che fino a un attimo prima erano vividi, ora sono coperti da una fitta nebbia che li rende confusi e offuscati. E tu non riesci a produrre neanche l'accento del meraviglioso discorso che ti eri immaginato. E resti pietrificato, fisso, a guardare il tuo passato, che ad andamento sostenuto (tipico dello scorrere del tempo) si allontana. In quel momento realizzi come qualsiasi tavola imbandita di svenevoli parole di cortesia sarebbe stata soltanto un inutile e pesante pasto di fronte ad un commensale già sazio.

Così, per rendere questa parte conclusiva della tesi più sopportabile, almeno dal mio punto di vista, la porterò avanti considerandola come un viaggio a ritroso nel tempo. Come un discorso fatto alla sola compagnia di me stesso.

Il mio percorso magistrale è stato, come ogni percorso che si rispetti, tortuoso e sterrato. Ma è stato proprio questo a renderlo tanto formativo. Se il me stesso di quando iniziai vedesse come sono diventato sarebbe contento? Forse. Se il me stesso di quando iniziai vedesse come sono cambiato sarebbe orgoglioso? Sicuramente. I profondi e imprevedibili eventi che hanno avuto luogo in questi 3 anni sono stati solo un assaggio di quello che è la vita. La vita vera. Quella cattiva, come una madre che sgrida il proprio figlio. Quella dolorosa, come lo schiaffo di un padre che vuole insegnarti una lezione. Quella appagante, come il panorama mozzafiato che scruti dopo aver scalato una montagna.

Buffo come riguardando indietro mi vengano in mente per lo più i momenti difficili, stampati con inchiostro indelebile nella mia corteccia celebrale. D'altronde non è forse vero che soltanto gli infanti e le bestie possono essere felici, poiché privi di memoria? Penso di sì.

Ed altrettanto buffo come solo ora mi renda conto che in questi momenti non ero mai solo. Me ne rendo conto proprio ora che sono solo. Strano. Tendo sempre a dare per scontato le cose positive, focalizzandoci sui traumi. Affascinante.

Vediamo allora se riesco a buttar giù tale scabroso e tetro muro di malinconia per edulcorare questo epilogo con i nomi di quelli che possono essere considerati i veri protagonisti di questa laurea, i miei angeli custodi, le mie cose positive.

Il primo pensiero è ovviamente rivolto al mio nucleo familiare. In particolare a mia madre, mio padre e mio fratello.

Mia madre, Porzia detta Lella, figura che nella mia microscopica esistenza è paragonabile a quella di una Dea, se la concezione di Dio è quella di Creatore. Mia madre mi ha creato, e come se non bastasse, non si è limitata a ciò. Ha continuato a curarsi di me senza pretendere nulla in cambio, anche nei momenti in cui me lo sono meritato meno. Ma in fondo la differenza tra amare e voler bene si può figurare nel rapporto di un uomo con un fiore. Se ti piace un fiore tu lo cogli, ma

quando ami un fiore continuerai a curarlo senza chiedere nulla in cambio. Ovviamente io sono il fiore, e grazie a lei la mia linfa vitale non si è ancora esaurita.

Mio padre, Aldo. Un uomo di poche ma buone parole. Un uomo mite che mi ha insegnato il rispetto ed il sacrificio. Un uomo di altri tempi, che in una società dove la coerenza ormai è un motivo di vergogna, mi ha tramandato ideali che contano più di qualche falsa amicizia in più. Un uomo che per me ha fatto più sacrifici di chiunque altro, e senza mai farmelo pesare né tantomeno notare. Ma io lo so, e farò tutto quel che è in mio potere per ripagarlo un giorno. Spero non troppo lontano.

Mio fratello, Alessandro. Un fratello come pochi ce ne sono. Penso che chiunque sarebbe, aggiungo giustamente, invidioso di un rapporto di fratellanza così ben riuscito. Un rapporto rafforzato da una complicità e sorprendente affinità nel modo di pensare e di approcciare la vita, che ci rende, a mio parere, la stessa persona in due corpi diversi. E chi meglio di te stesso con diverse esperienze può consigliarti? Questa serie di fattori non fa altro che saldare il nostro rapporto, pattuito oltre che dal corredo genetico, dall'affinità e stima reciproca. Sono sicuro sarà che ci saremo utili nelle nostre vite.

Precedente ho utilizzato il termine nucleo familiare. Scelta voluta, perché la mia famiglia non si limita ai consanguinei, ma è molto più allargata. Il mio pensiero va a loro, a uno a uno, gli devo molto, gli devo tutto. Ho deciso di non nominarne nessuno, perché in fondo ognuno di loro sa che ruolo ha giocato in questa incredibile partita durata più di qualche attimo. Gente con cui ho condiviso camere doppie, case durante la quarantena, pranzi, cene, telefonate infinite, viaggi, piste da ballo, festività lontano da casa, partitelle di calcetto, pomeriggi al campino, serate olandesi, serate torinesi, serate acquavivesi, weekend immersi nello sport, nottate di studio, aperitivi, bottiglie di vino, compleanni, progetti... e potrei continuare riempiendo intere pagine ma in questo periodo in cui la lotta ambientale è così importante non mi va di abbattere altri alberi. Insomma nelle risate e nei pianti la loro presenza non è mai stata in dubbio.

In questo percorso a ritroso non posso fare a meno di ripensare anche alle figure professionali che ho incontrato. Figure diverse, che mi hanno insegnato, che mi hanno supportato, che mi hanno illuminato. Colleghi, professori, professionisti che hanno reso il Politecnico una seconda casa per me. Anche grazie a loro devo la buona riuscita di questo percorso, oltre che di vita, è stato un percorso universitario. Colgo l'occasione per ringraziare in particolare il professor Carlo Novara, relatore di questa tesi, e il mio supervisore olandese Erjen Lefeber, una personalità fantastica che mi ha lasciato molto, non soltanto di nozioni tecniche ma anche di etica del lavoro, oltre ad esser stato il mio Cicerone nei Paesi Bassi, rendendo tale esperienza più piacevole.

Il viaggio in questi lunghi e brevi anni è terminato. Il mio pensiero durante le ultime righe di questo prologo si rivolge alla persona più importante della mia vita. La persona a cui penso prima di addormentarmi per addolcire il mondo onirico. La persona senza la quale sarei probabilmente molto peggio di quel che sono. Mia nonna, nonna Chella. Da bambino le dicevo che sarei diventato un mago per ringiovanirla e sposarla. Purtroppo nonna sono diventato un ingegnere, spero che questo ti renda comunque orgogliosa. Il mio "bagaglio della vita", come lo chiami tu, è evidentemente più pesante grazie a tutto ciò che ci hai messo dentro. Ti amerò per sempre.

Se avete letto tutta questa parte della tesi sappiate che probabilmente siete tra le persone che non ho citato, ma che fanno parte di me. Perché in fondo "in questo mondo terrificante, tutto quello che abbiamo sono i legami che creiamo".

Grazie.

Ivan Domenico Barbieri