ISSN 0280-5316
ISRN LUTFD2/TFRT--5707--SE

Dynamic Vision
Shape from Motion

Erik Casagrande

Department of Automatic Control
Lund Institute of Technology
June 2003

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
June 2003

Document Number
ISRN LUTFD2/TFRT--5707--SE

Author(s)
Erik Casagrande

Supervisor
Rolf Johansson LTH, Lund
Prof. Giorgio Picci Università di Padova, Italy

Sponsoring organization

Title and subtitle
Dynamic Vision from Motion (Dynamiskt Robotseende: Form från rörelse)

Abstract
Dynamic vision is a class of problems in computer vision that calculates properties of the3D world through the dynamics of the system observing by cameras. In this thesis we introduce the issue of motion and shape estimation with the aid of a single camera and for particular objects as points, planar surfaces and polyhedrons. This problem is well known in literature as Shape from Motion. We are interested in building a framework that, starting from a sequence of images that has been recorded from a camera, achieves the geometrical and dynamics properties of the 3D scene. In the first part we inspect the motion estimation through the 2D frames recorded by the camera. In particular we present a possible recursive approach for Optical Flow estimation. The second part introduces a new class of problems in system theory that are suitable for applications of computer vision. This subject has been called perspective system theory. We consider typical issue as observability, identifiability and realization theory for this branch of problems. Results has been shown through use of algebraic and recursive estimation algorithms.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
78

Recipient's notes

Security classification

Dynamic Vision
Shape from Motion

Erik Casagrande

8 July 2003

Contents

Chapter 1

Introduction

Dynamic Vision is a particular field inside of the Computer Vision that recovers information from a non-stationary sequence of images produced by cameras over the time. Thus the aim of this subject is to gain the information of the observed world from the perspective of motion, what is obtained either when the camera or object is moving. The acquired knowledge helps to understand the scene structure and the dynamics of the world. This field of study is well known also under the name *structure* or *shape from motion.*

The class of problems, hereby presented, has absorbed more and more attention for the last two decades. This fact is not surprising, since the computer vision is a challenging area of research for possible application in robotics and automation. Motion is fundamental to human vision. The fact that we humans understand the objects from a dynamic observation over the time is notorious from psychological studies. In contrary to human perception, for a robot the motion is the only clue and that is why it can sometimes drive in an unknown ambient. Thus in mobile robotics the information from a sequence of video frames ordered in time can be used to upgrade the navigation and control. Also the problem of calibration of the camera can be supported from the structure of the world observed over the time. However to improve dynamic vision field of study, some problems have to be clarified, what we try to contribute to, by the investigation described in my thesis.

1.1 Shape from motion

The term structure has two particular meanings: the static description of the world as well as the dynamic properties of the evolution over time.

The static part represents the geometry of the objects inside of the world. This assumption is connected to a position and orientation of simple geometrical entities, as for instance points, lines and planes. Inside of a real environment the class of objects can be bigger and more complex surface can be presented. In this case linear approximation has to be done to perform an easier description. If we look at the example of a sphere, it can be seen as the object covered by many tangent planes. These geometric characteristics, inside of this thesis, have been defined under the term *shape parameters*.

The dynamical properties are described by the velocities and the accelerations of the moving objects in the world. From the nature of the last characteristic is obvious that it is necessary to observe the world over the time with a number of camera frames grater than one. In case of the shape parameters a single image can give some clues of the scene geometry, but it does not provide a complete information. We present these characteristics with the name *motion parameters*.

More generally motion and shape parameters are described by the term *object characteristics*.

The camera within the configuration that we explained above, has the abilities to be an observer of the structure parameters of the particular piece of the world seen by it. Unfortunately, the camera model does not give a complete description of the structure parameters but instead it carries their equivalent representation that is given by *the image characteristic*. In fact, the camera was built to be a 2D observer for a 3D world and the transformation between the camera and the world is described by a function subjected to a *perspective projection*.

The function between image characteristic α_i and object characteristic β_i has been called *3D recovery equations* as it has been suggested in [2].

$$\beta_j = F_j(\alpha_1 ... \alpha_n) \qquad j = 1, ..., m \tag{1.1}$$

Unfortunately the 3D recovery functions from Eq (1.1) are non-linear and for this reason, it is difficult to attain the structure parameters.

1.2 Problem formulation

The purpose of this thesis is the implementation and the analysis of some topics of dynamic vision, to present and resolve the problem of structure from motion Moreover, we consider the problem of extracting the image characteristic trough a optical flow method.. The solution and the accomplishment of this goal, is related to the finding of an appropriate estimation scheme that is robust, efficient and suitable for real time applications.

The main set-up is built around a single camera observing the motion of simple objects and in particular planar surfaces. This assumption is obviously a strong constriction if we want to develop an algorithm inside of a real environment. The case of planar patch has been widely studied in the recent years and is well documented. For many reasons it is an optimal starting point for the motion estimation. Many human-made objects contain planar patches or at least they are modeled as easily as surfaces. Besides, as we have said previously, we can consider planar surface as a first order approximation of a complex surface. The case of a sphere covered by its tangent planes suits this concept.

This approach introduces immediately a limitation. The transformation between objects and image characteristics is not unique with one camera and it depends on a scale factor that is not possible to determine. Supporting the experiment with multiply camera system allows avoiding this problem, however its consideration is not the purpose of this thesis.

Within the described framework the motion of the objects have been studied through the case of *rigid body transformation*. It means that the motion does not deform the particular object undergoing the transformation, what is a typical situation in the real life.

Motion of the objects in the scene can be a consequence of some transformation applied on the objects themselves. The same motion can be produced from the movement of the camera in relation to a fixed environment. This particular case is known in the literature under the name *ego-motion*. In this way the estimation procedure regards just the motion parameters and the relative position of the camera with respect to the fixed background. Besides, the problems of moving camera and moving objects are dual problems linked to each other by a choice of a different reference point.

In case of robotics applications, the motion due to a camera movement is the main issue. A mobile robot or a robotic arm with a camera inserted are the examples of this case.

The computation is then divided into two major parts. The first one is addressed to the extraction of the image characteristic on the video stream and the second one concerns the estimation and the recovery of the structure parameters. Fig (1.2) shows a simple guideline of the process that has been studied.

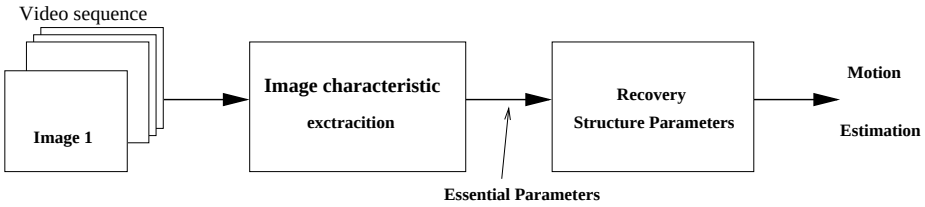


Figure 1.1: Framework base structure

Basically,there are two approaches that have been studied to extract the image characteristic from the video sequence. One method is to detect in a frame, particular geometric entities like points, lines curves. These entities are extracted from the brightness pattern and they are strictly connected to the objects present in the world, for instance the boundary of an object or the texture in a surface. The idea is then to track them and from the evolution of the trajectories recover the structure of the scene. Another approach, more general, of detection is to take care of every single point in a frame and to study the *optical flow* field. We address the estimation of the image characteristic in both directions but we inspect in details a particular approach for optical flow estimation. Motion of lines and points is simply a particular case of local optical flow. A big advantage of a dense computation as optical flow is the possibility to have a more general view on the system. Sometimes is also difficult to detect geometrical entities from a real scene. This solution, however, presents some drawbacks that most of the time brings the algorithm not to be so effective and realizable. A dense computation is slower and difficult to implement in real time. Another disadvantage is that the optical flow algorithms suffer from a lack of precision that creates a bad interpretation of the field.

The velocity description has to be supported by a stage of *segmentation*. In this way we can collect estimated data in order to understand which velocities are connected to the planar surfaces. The segmentation has also the duty to divide a region with different motion or different planes. This could

be done in order to attain a particular parametric model of the moving planar surfaces constrained by *essential parameters*. In general, the segmentation part is not an easy task and it should be carried out carefully. We are not interested in developing any segmentation algorithm. However, we can say that in order to achieve an efficient dynamic vision framework, this section has to be studied.

The last part concludes the framework through the interpretation of the estimated image characteristics. This is the kernel of the thesis and it has been studied with the auxilius of the new class of system theory called *perspective system theory*. The purpose is to find a robust estimator and to investigate the propriety of this new topic. We present our simulation by two approaches. One derives from a classical analytical method and aims to solve the 3D recovery equation. The second one is developed around an extended Kalman filter approach.

1.3 Background

The problem of shape from motion has been widely studied in Computer Vision and the literature that supports the topic is long.

The main source of knowledge which I based on in this thesis are the Ghosh papers. They introduce the problem of shape from motion through the class of perspective system theory [5][8][6] [7]. In this way the problem is centered around the extension of the well known system theory in order to study controllability, observability, realization theory etc... Mentioned papers refer to the particular analytical solution proposed by Kanatani [2].

Moreover, we consider the estimation of the image characteristic as an important step to gain a stable framework which solves our topics investigated within dynamic vision field. Inside the thesis we have paid a particular attention to the optical flow estimation. In [11] we found a general overview over the existent computation approach. It is not possible to list all the papers that has inspired the origin of this thesis. The basis of our work is included in [1]. In particular we are interested to find a recursive estimation that performs the algorithm over the time. In [12] the introduction of this kind of methods is developed together with a robust and recursive framework. In [3] we find a possible and easy approach that embodies the simple method presented in [1] and hereby, in my thesis, we take up a discussion of this exciting problem.

1.4 Tools

In present thesis we support the work with the aid of Matlab 12.1 and partially Simulink. Moreover, since we deal with computer vision algorithm we introduce two simulators to verify the approaches that have been studied in this thesis.

We created a first simulator into Matlab in order to have a better control of the noise apport and to test the correctness of the algorithms. Fig (1.2) is showing a synthetic 3D world system designed in our laboratory. This part, of course, can be done in Simulink, but we consider the visual tool important for understanding the behavior of our methods.

A second program that has been used is a 3D semi synthetic world built by an OpenGL engine. This program is the Virtual Robot developed by the Automatic Control Department at Lund University. A particular scene is presented in Fig (1.3).It has been observed by three different virtual cameras. We introduce some planar objects, for instance cubes, that help to carry out successfully our experiments. The cubes present different lifelike pattern in order to create a realistic situation.

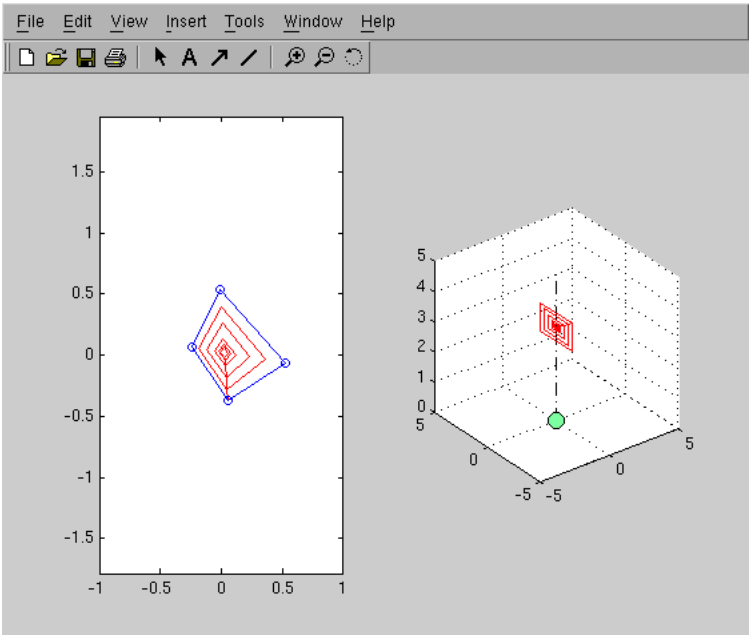


Figure 1.2: Synthetic 3D world used for simulation

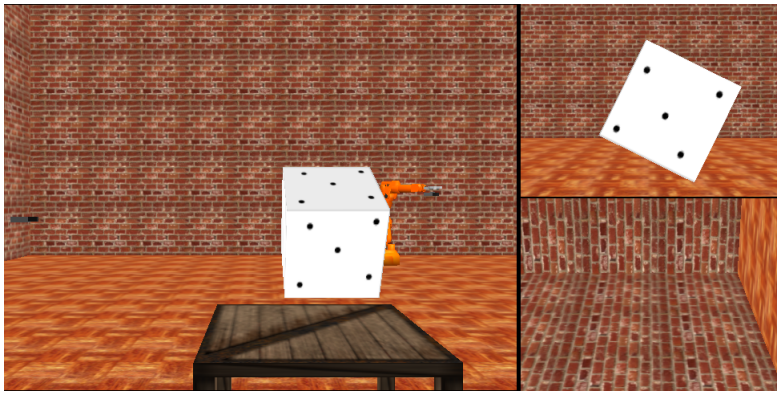


Figure 1.3: OpenGL world from the Virtual Robot Enviroment

We put a special emphasis on the last simulator to perform the studies of the optical flow algorithm.

The information captured by the virtual cameras were processed and stored in memory. This means that we did the work offline, because the algorithm is not optimized for the real time applications.

1.5 Overview of the thesis

We divide the thesis into following chapters:

In **chapter 2** the mathematical model of the framework of the thesis is analyzed. We start with investigation of the geometry of the camera model through different paths. Moreover, we consider a class of transformations described by the rigid body motion that introduces dynamics to the observed system. The specific case of moving planar surfaces has been studied.

In **chapter 3**, we consider the extraction of the image characteristics. Especially we focus on the estimation of the optical flow inside of a recursive algorithm. The purpose is to find an accurate dense field in order to be able to estimate the surface parametric model.

The **chapter 4** introduces some methods to resolve the shape from motion problem. In particular we have studied the perspective system theory that enlarges the classical field of system theory and thus embodies the specific cases of camera observations.

Chapter 2

Geometric Framework

In this chapter we present the mathematical model of the framework that has been studied in this work. We start with the evaluation of some possible models of camera from a geometric point of view and we describe the use of mentioned models inside of this thesis. The second part is centered on the study of the basic structure of the rigid body motion. We are interested, in particular, to describe the dynamics of the motion field associated with moving planar surfaces and in the formulation of the optical flow connected to them.

2.1 Camera models

The mechanism that is involved in the image formation in a typical camera follows the rules of a common optical system. The camera gets the light rays coming from the outside world. It comes through a lens, in the way to hit a screen inside of it. Typically the camera is digital and the screen is built with a sensor matrix (CCD). A simple situation is described in Fig (2.1) where a single ray has been projected behind the lens. The matrix screen is called usually *image plan.*

All the rays captured from the lens allow the formation of the picture of a particular part of the world space seen by the camera on the image plan.From a geometric point of view the system in Fig (2.1) can be described by the eq. (2.1). The parameter f inside of this equation is called *focal length* and it represents the distance from the lens to its focus.

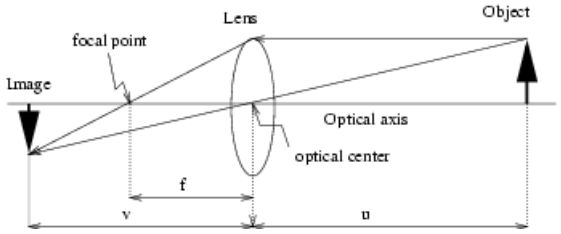


Figure 2.1: The basis of geometric optics

$$\frac{1}{v} + \frac{1}{u} = \frac{1}{f} \tag{2.1}$$

The system described is a simplification of what really happens inside of a camera. Much non-linearity has to be included to analyze correctly the model. A complete study of this optical system deals with the physical properties of the lens and to some extend with photometric knowledge. Those subjects are outside of the purpose of this thesis and we do not model our framework within this setup. Instead, we focus on a simplification of the real model that in most of the cases results in valid approximation.

A simple model that describes the behavior of camera and probably the most wildly used is the *pinhole* model. It describes an equivalent model from Fig (2.1) with a hole with infinitesimal diameter at the place of the lens. In this case we can consider that a single ray of light, produced by a single point in space, passes through the hole. The *features* on the image plan depend only on a single space point avoiding the complete studying of the image formation inside the photometric theory [1]. The pinhole model is described in Fig (2.2) and the law of the image formation that explains this model is called *perspective transformation*.

The model consists of a point O, called *center of projection* and the image plan. Distance f between them is called focal length, as described before. The plan is situated in front of the point O instead behind it, in order to simplify the model. The equation and the model of the camera does not suffer any changes except the fact that the coordinates are symmetrically reversed. At the point O, taken as origin, a coordinate system XYZ is changed, called *scene coordinate*. The axis Z is perpendicular to the image plan and is usually denoted as *Optical Axis*. At the image plan, instead, we associate another coordinate system xy called *image coordinates*. Now, we consider a point

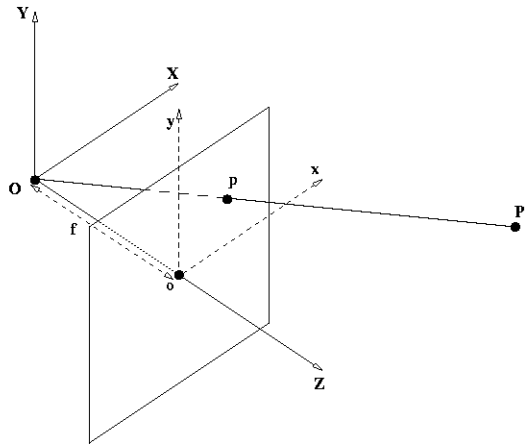


Figure 2.2: Perspective projection

P=[X,Y,Z] in the space described from scene coordinates. The line from P to O intersects the image plan in p=[x,y,z]. In this case z is always equal to f since the point belongs to the image plan. The coordinate x and y are expressed in image coordinate. The model can be described by the *projection equation* as follows:

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z} \tag{2.2}$$

An important particular is that the axis XYZ have been chosen in the way to have right hand system. For this reason we preferred to maintain the same letter on the image coordinate system and then, the x axis has a vertical orientation and the y axis horizontal orientation.

Model simplification

The simple model introduced above is clearly nonlinear. Inside of the computer vision community another structure called *Orthographic Projection* has been used for many applications. The new model comes from a relax version of the perspective transformation. In fact, if we consider the distance of two or more objects in the world as much bigger than the relative space between them, it is possible to approximate the parameter Z in Eq (2.2) with the average $\tilde{Z}$. Then the new transformation is given by

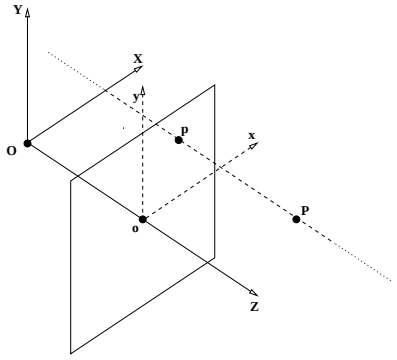


Figure 2.3: Orthographic projection

$$x = f\frac{X}{\widetilde{Z}} \qquad y = f\frac{Y}{\widetilde{Z}} \tag{2.3}$$

Moreover, assuming that the constant $\frac{f}{\widetilde{Z}}$ is unitary, we obtain finally:

$$x = X \qquad y = Y \tag{2.4}$$

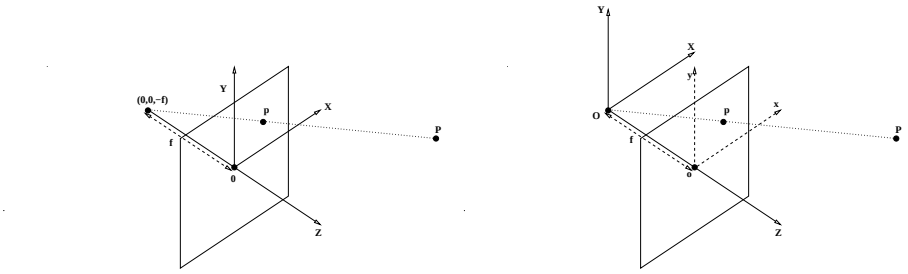
The model defined in the last eq. (2.4) describes a camera system where the rays that hit the image plan are perpendicular to each other, what is shown in Fig (2.3). This particular configuration is a bigger approximation than the perspective transformation of what really happened inside of a camera framework. This condition, substantially, holds when the object is compared with the focal length or is far away from the camera.

As described in [2][5] both projections above can be defined in a compact mathematical way. Let us introduce the following transformation:

$$x = f\frac{X}{Z+\delta} \qquad y = f\frac{Y}{Z+\delta} \tag{2.5}$$

where we add a parameter δ that can assume values in [0,f]. When we set $\delta = 0$, the transformation becomes identical with the formula described in the projective transformation in Eq (2.2). Using the term notation appears in [2], this geometric approach is defined as *viewer centered coordinate system* and the model based on that system has been described in Fig (2.2).

The formulation according with the setting $\delta = f$, expresses the camera system within the same perspective transformation but with the reference



(a) Image center (b) Viewer center

Figure 2.4: Coordinate System

point(0,0,0) shifted on the image plane to (0,0,f). This representation is called in [2] as *image centered coordinate system*. With the work progress we have found this approach more useful to inspect and to resolve analytically the problem of shape from motion through the 3D recovery equations as used in [2]. Another particular feature of the last coordinate system is that it has included the orthographic transformation described above. In fact if we take the limit of $f \to \infty$ we achieve the model shown in eq. (2.4).

2.2 Motion Analysis

In this section we consider that either the camera or the objects that compose the world are in movement. We introduce, for this reason, a class of linear transformations that are suitable to describe the dynamics of the rigid body motions. This group has the property that the distance between elements undergoing the same transformation does not change:

Let take a point in the world described by the three coordinates $\mathcal{X} = (X, Y, Z)$. We then consider a transformation on the 3D space into itself that presents the following motion dynamics

$$\dot{\mathcal{X}} = \mathbf{A}\mathcal{X} + \mathbf{b} \tag{2.6}$$

where we have

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \qquad \mathcal{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{2.7}$$

This linear transformation is defined by a generic 3x3 matrix in Eq (2.7) plus a vector (b_1, b_2, b_3). This model defines a motion called *affine*. In particular the values contained in the vector are the *translation velocities* of the specific point, which underwent the transformation.

The 3x3 matrix can be filled with arbitrary numbers but we restrict the values in the way to consider rigid body dynamics. In this case we would like to design the structure of the matrix to determine only rotation of points in respect to a reference axis. The set of matrices that describes rotation, forms a group denoted by SO(3) and is called *special orthogonal* [9] or *3D rotation group*. The matrices associated with this group take the special *skew symmetric* form, given by:

$$\mathbf{\Omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \tag{2.8}$$

The motion applying Eq (2.8) in Eq (2.6) describes an infinitesimal rotation around a reference point. It is completely defined by three velocity parameters inside of the vector $(\omega_1, \omega_2, \omega_3)$. This vector is called *rotation velocity*. By definition, the rotation vector delineates an axis of rotation passing through the reference point with angular velocity $\omega = \sqrt{\omega_1^2 + \omega_2^2 + \omega_3^2}$.

In this optics a rigid body motion is fully designed through six parameters of translation and rotation described above instead of twelve of the affine motion. This over-parametrization in the affine model determines a more general motion that also allows deformations of the objects, which underwent the transformation. In a real world, except few cases, it is enough to consider just six parameters mentioned above.

Through the thesis we prefer to describe the dynamics inside of an equivalent but more compact notation. In fact, using a new coordinate W we can homogenize the affine system in Eq (2.6). We get the following expressions:

$$X = \frac{\bar{X}}{\bar{W}} \qquad Y = \frac{\bar{Y}}{\bar{W}} \qquad Z = \frac{\bar{Z}}{\bar{W}} \tag{2.9}$$

According to the last equation and substituting into the affine model in Eq (2.6) we obtain the following equivalent system

$$\dot{\bar{\mathcal{X}}} = \mathcal{A}\bar{\mathcal{X}} \tag{2.10}$$

where

$$\bar{\mathcal{X}} = \left[\bar{X}, \bar{Y}, \bar{Z}, \bar{W}\right] \qquad \mathcal{A} = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 0 \end{pmatrix} \tag{2.11}$$

2.2.1 Discrete time analysis

We now, extend the consideration that we have performed for the continuous time to discrete time. This solution, in fact, is more suitable for computer operation.

$$\mathcal{X}_{k+1} = \mathbf{A}_d \mathcal{X}_k + \mathbf{g} \tag{2.12}$$

where $\mathbf{A}_d$ and $\mathbf{g}$ are, respectively, the equivalent of discrete 3x3 matrix rotation and the translation vector. In particular, when the continuous rotation matrix is skew-symmetric as in Eq (2.8), the matrix transformation to the discrete form takes an easy calculation given by

$$\mathbf{A}_d = e^{\mathbf{\Omega} T}, \qquad \mathbf{g} = \int_0^T e^{\mathbf{\Omega}(T-\tau)} \mathbf{t} d\tau \tag{2.13}$$

where we define T as the sampling time. The matrix, due to the previous transformation, $\mathbf{A}_d$ does not have any particular symmetric form as in Eq (2.8) .

In conclusion, the introduction of the homogeneous coordinates for the discrete time produces the same equation that in Eq (2.10) merely with the changes in the term of the matrix $\mathcal{A}$ substituted by 1

$$\begin{pmatrix} \mathbf{A}_d & \mathbf{g} \\ 0 & 1 \end{pmatrix} \tag{2.14}$$

2.3 Moving planar surfaces

Moving planar patches can be described as the motion field of the points attached to that surface. All of these points during the motion undergo the

rigid body transformation that has been expressed in Eq (2.6). Subjected to the planarity condition, the relationship between points on the surface is constraint by the following geometrical expression:

$$Z = pX + qY + r \tag{2.15}$$

This equation on the 3D points contains three parameters (p,q,r) that represent the position and the orientation of the planar patch in the world. The parameters have been called *shape* of the planar surface. The couple *(p,q)*, in particular, indicates the component of the normal vector on the surface and next it defines the plan orientation. The parameter r is the distance from the coordinate reference point of the world to the plan in terms of the coordinate Z and it has been called *absolute depth.*

Now we assume to have a camera that observes the motion of this surface on the 3D world through a rigid body motion described above. The point where the transformation has been applied is the intersection between the plan and the Z axis. This motion is simply described by a rotation $(\omega_1, \omega_2, \omega_3)$ along an axis and a vector translation given by (b_1, b_2, b_3). The situation has been shown in Fig (2.5) together with the camera model described in this figure by the pinhole image centered coordinate system.

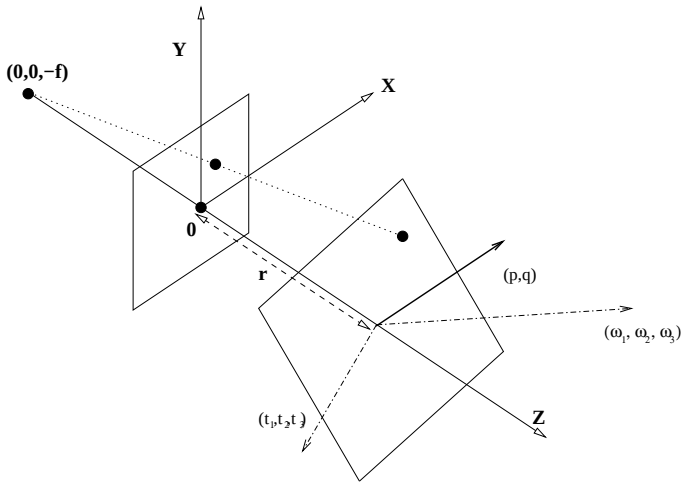


Figure 2.5: Planar Rigid Motion

The observation of each single point of the camera that is subjected to the rigid body transformation, produces an equivalent motion field via the

perspective of the camera on the image plan. This motion field is the optical flow that we are interested in and on which we based the process of estimation. The following preposition that has been taken from [2], expresses the particular mathematical structure of the optical flow of the surface subjected to the motion and to the chosen camera.

Proposition 2.1 *The optical flow induced by perspective projection of planar surface motion is given by:*

$$\begin{aligned}\dot{x} &= d_1 + d_3 x + d_4 y + \tfrac{1}{f}(d_7 x^2 + d_8 xy)\\ \dot{y} &= d_2 + d_5 x + d_6 y + \tfrac{1}{f}(d_7 xy + d_8 y^2)\end{aligned}$$

where

$$\begin{aligned} d_1 &= f\tfrac{b_1}{\delta+r} & d_5 &= \omega_3 - p\omega_1 - \tfrac{pb_2}{\delta+r}\\ d_2 &= f\tfrac{b_2}{\delta+r} & d_6 &= -q\omega_1 - \tfrac{qb_2+c}{\delta+r}\\ d_3 &= p\omega_2 - \tfrac{pb_1+c}{\delta+r} & d_7 &= \tfrac{1}{f}\left(\omega_2 + \tfrac{pc}{\delta+r}\right)\\ d_4 &= q\omega_2 - \omega_3 - \tfrac{qb_1}{\delta+r} & d_8 &= \tfrac{1}{f}\left(\omega_1 + \tfrac{qc}{\delta+r}\right)\end{aligned} \tag{2.16}$$

The optical flow for a moving planar surface via perspective projection is defined completely by eight d parameters introduced above. They only contain information about the shape of the plan and the motion transformation applied. These parameters are called *flow* [2] or *essential* [5] *parameters*. They are the goal for the motion estimation on the image plan.

2.3.1 Shape dynamics

The motion acts on the plan influencing the shape parameters over the time. We can calculate this relationship from the motion dynamics of a single 3D point in Eq (2.10) together with the planar constraint in Eq (2.15). Introducing homogeneous coordinates for the shape parameters we obtain a compact matrix notation, given by:

$$\frac{d}{dt}\left(\bar{p}, \bar{q}, -\bar{s}, \bar{r}\right)^T = \mathcal{A}\left(\bar{p}, \bar{q}, -\bar{s}, \bar{r}\right)^T \tag{2.17}$$

where

$$-\mathcal{A}^T = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 0 \end{pmatrix} \tag{2.18}$$

and the shape vector is homogenized as:

$$p = \bar{p}/\bar{s} \qquad q = \bar{q}/\bar{s} \qquad r = \bar{r}/\bar{s} \tag{2.19}$$

Considering in Eq (4.12), the special case of rigid body motion described previously, the matrix $\mathbf{A}$ has to be changed to the skew-symmetric counterpart Ω defined in Eq (2.8). This is given by:

$$\frac{d}{dt}\begin{pmatrix} \bar{p} \\ \bar{q} \\ -\bar{s} \\ \bar{r} \end{pmatrix} = \begin{pmatrix} \mathbf{\Omega} & 0 \\ -\mathbf{b}^T & 0 \end{pmatrix} \begin{pmatrix} \bar{p} \\ \bar{q} \\ -\bar{s} \\ \bar{r} \end{pmatrix} \tag{2.20}$$

The last dynamic equation describes the behavior of the time varying shape parameters over the rigid transformation. This relationship is a non-linear function on the shape coefficients. In fact from the Eq (2.20), if we rewrite the system in non homogeneous coordinates we attain the second order differential equation as follows:

$$\begin{aligned} \dot{p} &= \omega_1 q - \omega_2(1+p^2) - \omega_3 pq \\ \dot{q} &= -\omega_1 p - \omega_3 - \omega_2 pq - \omega_3 p^2 \\ \dot{r} &= -(\omega_3 q + \omega_2 p)r + (b_3 - b_2 q - b_1 p) \end{aligned} \tag{2.21}$$

This equation is a differential Riccati equation. For this reason we can refer the Eq (2.20) as the *Riccati dynamics*.

2.3.2 Discrete shape dynamics

The analysis that has been done for the shape dynamics in the previous section, has to be extended for an easy implementation in a recursive algorithm suitable for computer applications. The situation that we consider is the moving planar surface with the following discrete constrain equation:

$$Z_k = p_k X_k + q_k Y_k + r_k \tag{2.22}$$

We begin to consider the flow equation for the perspective camera in discrete time for the feature (x_k, y_k) on the image plan. The camera model is the viewer centered coordinate model.

Thus is given by:

$$\begin{aligned} x_{k+1} &= \frac{d_1 x_k + d_2 y_k + d_3}{d_7 x_k + d_8 y_k + d_9} \\ y_{k+1} &= \frac{d_4 x_k + d_5 y_k + d_6}{d_7 x_k + d_8 y_k + d_9} \end{aligned} \tag{2.23}$$

where the discrete essential parameters are:

$$\begin{aligned} d_1 &= a_{11} - c_1 p_k \quad d_2 = a_{12} - c_1 q_k \quad d_3 = a_{13} - c_1 \\ d_4 &= a_{21} - c_2 p_k \quad d_5 = a_{22} - c_2 q_k \quad d_6 = a_{23} - c_2 \\ d_7 &= a_{31} - c_3 p_k \quad d_8 = a_{32} - c_2 q_k \quad d_9 = a_{33} \end{aligned} \tag{2.24}$$

and the relative translation velocities are:

$$\begin{aligned} c_1 &= \frac{g_1 - a_{13}\delta}{r_k + \delta} \\ c_2 &= \frac{g_2 - a_{23}\delta}{r_k + \delta} \\ c_3 &= \frac{g_2 - (a_{33} - 1)\delta}{r_k + \delta} \end{aligned} \tag{2.25}$$

Inside of the previous equation, we use the dynamical discrete model described in Eq (2.12). The terms g_j represents the discrete translational velocity while the a_{ij} constitute the matrix $\mathbf{A}_d$.

Finally, it can be useful to rewrite the Riccati dynamics presented in the continuous time analysis, what is given by:

$$\begin{aligned} p_{k+1} &= \frac{\Delta_1 x_k + \Delta_2 y_k + \Delta_3}{\Delta_7 x_k + \Delta_8 y_k + \Delta_9} \\ q_{k+1} &= \frac{\Delta_4 x_k + \Delta_5 y_k + \Delta_6}{\Delta_7 x_k + \Delta_8 y_k + \Delta_9} \end{aligned} \tag{2.26}$$

where the including terms Δ_{ij}, are the cofactors of the discrete matrix $\mathbf{A}$ in Eq 2.12).

2.3.3 Other models

The image centered coordinate system includes the possibility to take the limit $f \to \infty$ inside the projective transformation formula in order to achieve the orthographic camera model. The same operation can be used to calculate the orthographic flow equation. In fact, considering the eight essential parameters above and taking the limit $f \to \infty$ we conclude the following result:

Proposition 2.2 *The optical flow induced by orthographic projection of planar surface motion is given by*

$$\dot{x} = d_1 + d_3 x + d_4 y$$
$$\dot{y} = d_2 + d_5 x + d_6 y$$

where

$$\begin{array}{ll} d_1 = b_1 & d_4 = -\omega_3 + p\omega_2 \\ d_2 = b_2 & d_5 = \omega_3 - p\omega_1 \\ d_3 = p\omega_2 & d_6 = -q\omega_1 \end{array} \tag{2.27}$$

The previous model via orthographic transformation contains only six parameters. This model of optical flow has been widely used and it has been called *affine model*. The advantage of this model is the reduction of the parameters number that has to be detected and thus a quick estimation procedure. Beside the last two parameters of the eight coefficient model contain a large amount of errors and it is difficult to make a correct estimation of them.

To complete the analysis of the optical flow model we would like to mention another kind of model described in [2]. It's called *Pseudo-Orthographic model approximation* and it contains features between the perspective an orthographic system. In fact if we consider the focal length $f \to \infty$, but not big enough to consider a valid orthographic model, we can omit the term $O(\frac{1}{f^2})$ in the essential parameters in Eq (2.16) and instead keep the terms $O(\frac{1}{f})$. This operation influences the last two parameters of the flow. The model is then given by:

Proposition 2.3 (Kanatani) *The optical flow induced by perspective projection of planar surface motion is given by:*

$$\dot{x} = d_1 + d_3 x + d_4 y + \tfrac{1}{f}(d_7 x^2 + d_8 xy)$$
$$\dot{y} = d_2 + d_5 x + d_6 y + \tfrac{1}{f}(d_7 xy + d_8 y^2)$$

where

$$\begin{array}{ll} d_1 = f\frac{b_1}{\delta+r} & d_5 = \omega_3 - p\omega_1 - \frac{pb_2}{\delta+r} \\ d_2 = f\frac{b_2}{\delta+r} & d_6 = -q\omega_1 - \frac{qb_2 y+c}{\delta+r} \\ d_3 = p\omega_2 - \frac{pb_1+c}{\delta+r} & d_7 = \frac{1}{f}\omega_2 \\ d_4 = q\omega_2 - \omega_3 - \frac{qb_1}{\delta+r} & d_8 = \frac{1}{f}\omega_1 \end{array} \tag{2.28}$$

Chapter 3

Image characteristic estimation

In the previous chapters, points in the 3D space have been calculated and subjected to a rigid body transformation. This motion, instantaneously, produces a *motion field* where each 3D vector has a meaning of velocity of the points in the space. The projection of this field onto the image plan of a camera shows a similar result but in a 2D space. This representation of the rigid motion is usually called *optical flow* as described in many books of computer vision. It is a reliable model to describe the velocity produced either with the camera motion (*ego-motion*) or moving body in the space with a static camera. However, as it has been pointed out by many authors, it is more correct to describe it as the motion of the brightness pattern on the image sequence. In fact if we take a surface without any pattern drawn on it, the camera cannot detect any movement and in that particular region, the optical flow result is null everywhere. For this reason, after having obtained an image frame, the locations where the motion field can be possibly detected, are where there is discontinuity in the brightness pattern.

The strategies to solve this problem of motion estimation in the image plan can be divided into two categories:

Dense computation that presupposes the estimation of the velocity field for each single pixel in the image. This technique is referred as *optical flow estimation* to underline the complete evaluation of the field.

Sparse computation where the main purpose is to identify particular feature on the image and calculate the velocity for this entity. The features can be single points, lines, curves etc. The process usually contains an algorithm of *tracking and matching feature* over the time.

These two problems are of course connected to each other for the simple reason that the motion of features is a local estimation of the dense computation. The whole optical flow, since it has been calculated at each pixel, it embodies more information about the dynamics of the observed system. Unfortunately, the drawback of this technique is the dramatic load of complexity calculation that does not give an attractive face to this approach. We would like, in the sequel, to inspect the problem of dense estimation of optical flow and to determine some characteristic that would be suitable within a dynamic vision framework. Both dense and sparse approaches are well documented in the literature, i.e. [10] [1] and in many other book concerning computer vision.

Framework image characteristic estimation

For the purpose of this chapter and the thesis we are interested in the computation of the flow field for the particular problem of moving planar surface. The target is to support the problem structure from motion by evaluating, as we said in the introduction the image characteristics. For our duty, the motion of planar surfaces is described completely from the essential parameters in Prep (2.1) and they are the goal of the process evaluation. This explanation can be summarized in the estimation framework described by the sketch in Fig (3.1).

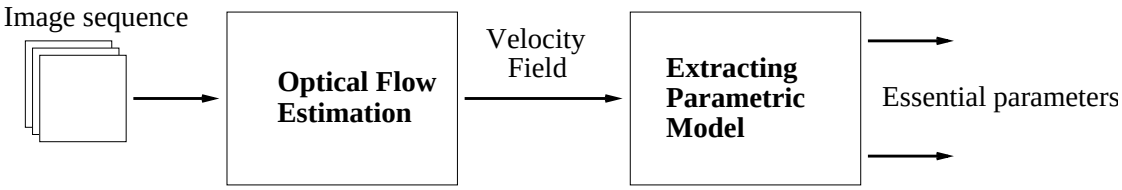


Figure 3.1: Optical Flow scheme estimation

Basically, we need a stage for estimating the optical flow either in a dense or in a sparse computation. Inside, we include a pre-stage that contains basic image processing algorithm prepared for an optimal flow detection. It includes basic noise suppression, image filtering and edge detection. The second stage implements an algorithm to detect the essential parameters that we need in the next chapter.

3.1 Feature estimation

The estimation of the optical flow through features on the image plan is the most common for an easy use and application. As we described before it consists of two phases.

The first one concerns the detection of particular features on the images as points, lines or complex curves. The case of simple points has been studying quite a lot in the Computer vision community. In my thesis the use of the Virtual environment can help not to take under consideration a complex algorithm for real environment. Fig (3.2) shows a simple cube with an abstract pattern including some points. Our algorithm simply thresholds the image and 5 regions of pixels that belong to the points. The next step is to find the centroid of the regions in order to extract a single feature.

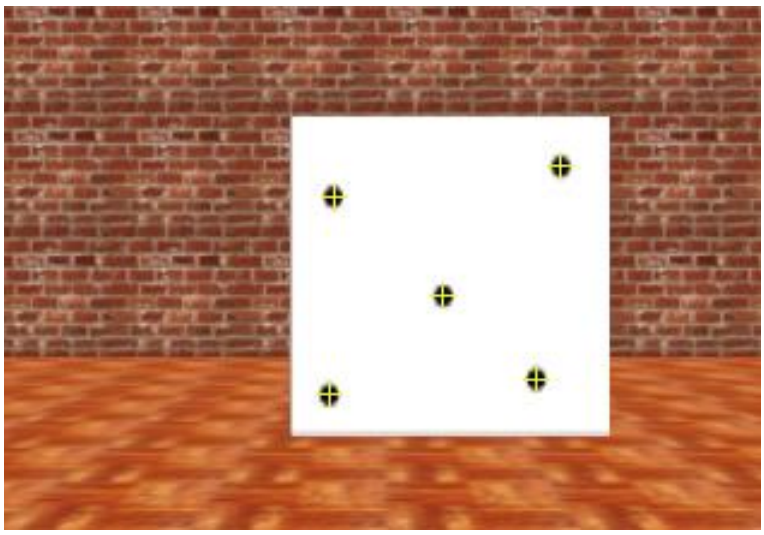


Figure 3.2: Feature example

With the availability of the sequence of images and the localization of feature, the second part has the duty to provide the correct match of features between a pair of frames. In this way, I can determine its history, initial point and evolution to the final point. This procedure is called *Tracking*. In the next chapter we include a Kalman filter approach to track the features on the image sequence. This is the typical solution for tracking. Moreover the

algorithm performs the following of the features together with the shape from motion estimation.

3.2 Optical Flow field estimation

.

There are many computational techniques to evaluate the velocity field. They can be divided into following groups: *(1)Gradient-based methods*, which compute optical flow using information about spatial derivatives of the images, *(2)Correlation-based methods*, which perform a matching strategy to search the parts that are most correlated to each other and *(3)Frequency-based methods* which typically use filtering approach to operate on the phase propriety of the flow. An introduction and an overview of those algorithms is presented in [11].

In this thesis the estimation of the flow is calculated using the basic method belonging to the gradient-algorithm class. This approach is based on the famous Horn method in [1] that includes some basic concepts about the optical flow estimation. In the second stage we would like to integrate Horn algorithm building a recursive framework that has been proposed in [3].

3.2.1 Brightness equation

It is possible to assume that between two frames in a time-ordered sequence the displacement of the features is rather little. Subjected to this fact, the brightness pattern of a region is approximately constant under the motion between these frames. This means that in a small region, the intensity function of the brightness pattern remains the same even if the location can change over time.

Let the image intensity function *I(x,y,t)* with *[x,y]* a particular point at time t. It is possible to express the data constraint as follows:

$$I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t) = I(x+u\delta t, y+v\delta t, t+\delta t) \qquad (3.1)$$

where δx ,δy and δt are the displacement between two images and $\mathbf{v} = [u,v]$ is the instantaneous velocity under the rigid motion, which has to be found. As described before, the sense of Eq (3.1) locally states that intensity

is unchanged over a short time period. With a reasonable assumption that the brightness varies smoothly over the spatio-temporal coordinates, the left-hand side of the equation above can be expanded using a first order of Taylor series:

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x} u \delta t + \frac{\partial I}{\partial y} v \delta t + \frac{\partial I}{\partial t} \delta t + O^2 \tag{3.2}$$

The last terms O^2 are negligible because terms of second and higher order. They will not be taken under consideration for the computation.

Finally we rewrite the equation above in a standard form for the constrain.

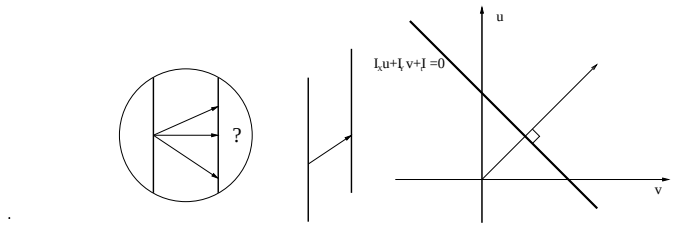
$$I_x u + I_y v + I_t = 0 \tag{3.3}$$

This formula is the basic for most of optical flow algorithms and is referred as *optical flow constant equation* [1] or *Brightness constraint.* From this equation using, for instance, a simple least-squares approach we can calculate the vector $\mathbf{v}$. However the constraint presents a weakness and the problem is ill-posed. In fact, the constraint presents one equation in two unknowns per every pixel. The formula allows to recover the velocity vectors only in the direction of the spatial gradients I_x and I_y. The Fig (3.3(b)) describes this phenomenon in the u-v plane.

$$\mathbf{v}_\perp = \frac{-I_t \nabla I}{\|\nabla I\|_2^2} \tag{3.4}$$

The physical explanation of this problem has been shown in Fig (3.3(a)). The motion and the optical flow is easily detected locally but inside of a global view may be incorrect. We cannot compute the real optical flow from Eq (3.3) because the recovery of the motion can be ambiguous.

This situation is usually referred in the literature as *the aperture problem.* The flow, then, has to be recovered, operating not only locally, around one pixel as the brightness equation describes, but doing the computation in a larger region. In this case the problem can be resolved, but the validation of the flow equation and then the assumption of brightness equation cannot be valid any longer.



(a) Unknown directions (b) Constraint equation

Figure 3.3: Aperture Problem

3.2.2 Smoothness constraint

The brightness equation above describes partially the optical field and it produces a set of infinitive possible solutions. This method is then ill-poseness and it suffers from the well known aperture problem. It is necessary to restrict the number of solution adding a regularization term in the computation of the optical flow. This new constraint is based on the assumption of the *spatial coherence*. The motion in a real environment does not influence one single pixel but, likely, one extended region. In fact, the real objects have a spatial extension of points connecting together. Under this optics, the flow estimation on a particular pixel is related to its neighborhood field. This means that the optical flow varies smoothly in some region of the image. The study of the local interference between pixels is helpful to build an extra set of constraints.

In the Horn method, the minimization problem, partially described by the brightness constraint, includes a new regularization term and is given by:

$$\min_{u,v} E(u,v) = E_{brightness}(u,v) + \lambda E_{spatialcoherence}(u,v) \qquad (3.5)$$

The term lambda controls the relative weights between the two constraints. This formula restricts the minimization problem in order to achieve one single solution. It helps the optical flow algorithm to become well-posed and suitable for being used.

The most common approach to constrain locally the field is the formu-

lation through the *membrane model*. It can also frequently called under the name *first order model*. The regularization term that has been introduced before, takes the following form:

$$E_S = u_x^2 + u_y^2 + v_x^2 + v_y^2 \tag{3.6}$$

The constraints smoothes the flow through the relationship between the derivatives of the velocities in a particular point of the image. To be suitable for computation we design this term in an another form. Since this image is a discrete map, it is convenient to describe it as a grid of pixels. The Fig (3.4) shows this grid locally in a neighborhood of the point (i,j). The formula inside of Eq (3.6) can be rewritten as the relationship between the cardinal pixels of (i,j).

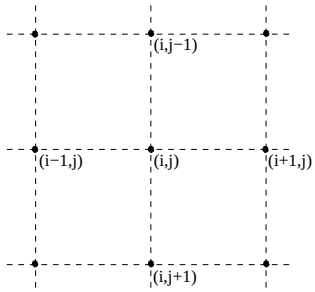


Figure 3.4: Pixels grid

In this way we have from Eq (3.6):

$$E_s = \frac{1}{2}\left[(u_{i,j} - \bar{u}_{i,j})^2 + (v_{i,j} - \bar{v}_{i,j})^2\right] \tag{3.7}$$

where the symbols $\bar{u}$ and $\bar{v}$ represent the average flow at the cardinal pixels.

This smoothness operation is described in as a simple convolution between the kernel in Eq (3.8) and the component of the velocity field matrix. Thus, we can give manually the weight of importance to the neighborhood pixel.

$$h = \frac{1}{20}\begin{bmatrix} 1 & 4 & 1 \\ 4 & 0 & 4 \\ 1 & 4 & 1 \end{bmatrix} \tag{3.8}$$

This form is adopted in the next section to implement the recursive algorithm.

Another model suitable as a constraint is the *second order or affine model.* The description of this solution is described in [13], but in the following operation will not be used. The formula of the constraint takes care of the second flow derivatives and is described by:

$$E_S = u_{xx}^2 + u_{yy}^2 + 2v_{xy}^2 \tag{3.9}$$

The solution of the minimization problem in Eq (3.5) is performed in [1] as an example of calculus of variation. Avoiding the computational steps to get this equation solved, the solution has been computed through an iterative method like Gauss-Siegel. In fact in [1] the following algorithm is presented:

$$u_{i,j}^{n+1} = \bar{u}_{i,j}^{n} - \frac{I_x \left(I_x \bar{u}_{i,j}^{n} + I_y \bar{v}_{i,j}^{n} + I_t \right)}{1 + \lambda \left(I_x^2 + I_y^2 \right)} \tag{3.10}$$

$$v_{i,j}^{n+1} = \bar{v}_{i,j}^{n} - \frac{I_y \left(I_x \bar{u}_{i,j}^{n} + I_y \bar{v}_{i,j}^{n} + I_t \right)}{1 + \lambda \left(I_x^2 + I_y^2 \right)} \tag{3.11}$$

The computation is refined for every iterations within the difference between two frames.

The assumption of spatial coherence is a powerful tool to constraint the brightness equation but it suffers from some limitations. The major problem is that the application of the spatial coherence across the boundary of moving surface may give wrong results. In fact, it is typical in those zones to compose the reciprocal relation for different kind of motion models through the smoothness constraint. A robust computation has to detect those areas and disconnect the pixels belonging to different motion models. Moreover, to obtain a better performance of the estimation process, one can think to increase the neighborhood of a pixel (i,j). Particular attention has to be paid, because the larger the region, the bigger the probability to get the motion boundary problem.

3.3 Recursive adaptive estimation

The method presented in [1] describes an algorithm using spatio-temporal derivatives to achieve the velocity field between two singular frames. The

Gauss method performs a global approximation of the real flow, within a high number of iterations. The accuracy of the flow in many cases depends strongly on the iterations and then the computational time is really expensive. This is a real problem if the goal of using optical flow is centered inside of real time system and application. The case of study encountered in this thesis is, among others, focusing on that issue. For particular application in image processing, for example, video decoding and encoding, where the optical flow is a field of study, the developed frameworks belong to the batch programming. The time taken for this kind of application is less important than the precision. Instead, if we think about a robot that implements an algorithm of structure from motion in order to achieve real time information about the ambient where it has been placed, we can understand the importance of a low time complexity. It is necessary to include an algorithm that performs the computation on a longer sequence of image over the time, in order to increase the speed and to get better performance. Introducing the recursive fashion framework fulfills this important characteristic. Moreover, it is important to underline that the calculus based on more than two frame includes the propriety of integral computation. This means less sensitivity and more accurate detection of the real flow. At the end, another preferable characteristic is that a recursive algorithm is adaptive by nature. The parameters inside of the optical flow equation change over the time getting a reasonable accuracy. These reasons explain why in the recent years more attention has been paid on the development of recursive estimation frameworks.

The approach adopted in [3] describes a technique which adds a temporal constraint in the Horn approach presented before. The idea is to achieve a low-computational cost algorithm, which would also be easier to implement. The adaptive and recursive characteristics based on that assumption, fulfill the initial idea supported in this thesis.

We begin with putting the matrix notation of the spatio-temporal gradient of the video sequence using the notation in [3]:

$$\mathbf{Y}(t) = -\left[\frac{\partial I(1,1,t)}{\partial t} \cdots \frac{\partial I(x,y,t)}{\partial t} \cdots \frac{\partial I(N,N,t)}{\partial t} \right] \tag{3.12}$$

$$\mathbf{H}(t) = -\left[\begin{array}{ccc|ccc} \frac{\partial I(1,1,t)}{\partial x} & & 0 & \frac{\partial I(1,1,t)}{\partial y} & & 0 \\ & \ddots & & & \ddots & \\ 0 & & \frac{\partial I(N,N,0)}{\partial x} & 0 & & \frac{\partial I(N,N,t)}{\partial y} \end{array}\right] \tag{3.13}$$

$$\mathbf{X}(t) = \begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \end{bmatrix} \text{ where } \begin{matrix} \mathbf{D}_x = \begin{bmatrix} \frac{\partial I(1,1,t)}{\partial x} \cdots \frac{\partial I(x,y,t)}{\partial x} \cdots \frac{\partial I(N,N,t)}{\partial x} \end{bmatrix} \\ \mathbf{D}_y = \begin{bmatrix} \frac{\partial I(1,1,t)}{\partial y} \cdots \frac{\partial I(x,y,t)}{\partial y} \cdots \frac{\partial I(N,N,t)}{\partial y} \end{bmatrix} \end{matrix} \quad (3.14)$$

By combining the matrix notations in above equations and the algorithm proposed by Horn, we obtain the brightness equation and consequently the spatial coherence assumption. The description is completed by introducing an error model in the brightness equation. This term includes possible sources of noise, and is given by:

$$\mathbf{Y}(t) = \mathbf{H}(t)\mathbf{X}(t) + \mathbf{E}(t) \qquad with \qquad E\left\{\mathbf{E}(t)\mathbf{E}^T(t)\right\} = \sigma_e^2(t)I \quad (3.15)$$

$$\left|\tilde{\mathbf{S}}\mathbf{D}_x\right|_2^2 + \left|\tilde{\mathbf{S}}\mathbf{D}_x\right|_2^2 = \left|\mathbf{S}\mathbf{X}(t)\right|_2^2 \quad (3.16)$$

Inside the equation Eq (3.16), the matrix S includes the relationship between feature pixels in order to describe the spatial coherence of adopted model. In fact, the matrix is designed to have a spatial relationship between neighborhood pixels similar to the one described before in Eq (3.8). For this reason we have the special diagonal structure of matrix S. Following the denotation in Eq (3.16) two small matrices $\tilde{\mathbf{S}}$ have for each line 9 parameters from Eq (3.8). In fact each line represents the relationship between pixels. The matrix S is then structured as:

$$\left|\begin{array}{c|c} \tilde{\mathbf{S}}\tilde{\mathbf{S}} & 0 \\ \hline 0 & \tilde{\mathbf{S}}\tilde{\mathbf{S}} \end{array}\right| \quad (3.17)$$

The description adopted in Eq (3.12) can be used around the algorithm of Horn to formulate a matrix minimization problem equivalent to Eq (3.5). To add a temporal refinement to this algorithm is more suitable to build a state space theory:

$$\begin{bmatrix} \mathbf{Y}(t) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{H}(t) \\ \mathbf{S} \end{bmatrix} \mathbf{X}(t) + \begin{bmatrix} \mathbf{E}(t) \\ \mathbf{F}(t) \end{bmatrix} \quad (3.18)$$

where the model of the noise is described by:

$$\mathbf{E}(t) = N(0, \sigma_e) \qquad and \qquad \mathbf{F}(t) = N(0, \sigma_f) \quad (3.19)$$

This state space formulation can be resolved using traditional recursive technique. For instance, the Kalman filter is well-known to be the optimal

solution if we used the correct noise model. This theory is well developed and the simple implementation of the computation is an advantage that influences the choice of this algorithm. Moreover, the application of Kalman filter has an extension use inside of computer vision for tracking, for instance image features. In our optical flow estimation, instead, the implementation of it is built around a dense computational framework. In case of Kalman filtering suffers from a dramatic computational cost. The dimension of the matrix involved in the formulation in Eq (3.16) results in prohibitive computation for the fact that are expensive matrix inversions. In our particular case, another disadvantage is the amount of memory indispensable for this kind of approach. The matrix in Eq (3.16) is sparse, indeed.

The key of idea to perform temporal refinement is the use of a weight least square approach over the time. Referring the whole calculation to [3], in order to minimize the formulation of the squared error in Horn's algorithm, it has to be changed to include a weighting factor. From this point of view the solution of the problem can be centered inside of a recursive least square estimation. The error is given by:

$$\epsilon^2(t) = \frac{1}{\sigma_e^2} \sum_{k=0}^{t} \phi(t,k) \left(|\mathbf{Y}(t) - \mathbf{H}(t)\mathbf{X}(t)|_2^2 + \beta \, |\mathbf{R}\mathbf{X}(t)|_2^2 \right) \tag{3.20}$$

where the term is the exponential weight and it is described by:

$$\phi(t,k) = \begin{cases} 1 & k = 0 \\ \prod_{i=0}^{k-1} \lambda(t-i) & k \geq 1 \end{cases} \tag{3.21}$$

The parameter $\beta = \frac{\sigma_e^2}{\sigma_f^2}$ defines the smoothness error. The term λ act as forgetting factor of this algorithm. The purpose is to rise the variance of the model exponentially, when the data is taken far away from the current estimator. The Eq (3.20) attains the minimum value when the derivative of the error is equal to zero.

$$\frac{\partial \epsilon^2(t)}{\partial \mathbf{X}(t)} = \frac{2}{\sigma_e^2} \left[\mathbf{R}(t)\mathbf{X}(t) - \mathbf{P}(t) \right] = 0 \tag{3.22}$$

The matrix R functions as the *covariance* matrix of the filter presented in Eq (3.18) while the matrix P, for the same reason, is the *cross-correlation* matrix between the input and the unknown X(t). Following some further

computations or having a look into the theory of WLS, [4], these two matrices are recursively calculated as:

$$\mathbf{R}(t) = \delta(t)\mathbf{R}(t-1) + \mathbf{H}^T\mathbf{H} + \beta\mathbf{S}^T\mathbf{S} \quad (3.23)$$

$$\mathbf{P}(t) = \delta(t)\mathbf{P}(t-1) + \mathbf{H}^T\mathbf{Y} \quad (3.24)$$

The achievement of the adaptation in the temporal domain is then regarded as a recursive least square fashion algorithm. Unfortunately, for the same reason as in case of the Kalman filter, this approach suffers from the dramatical computation problem, which is the inversion of the sparse and huge matrix. That is why this method is not suitable for online programming. The authors in [3] present a strategy similar to the recursive least square but without performing any matrix inversion. The error is found using the directly the solution from Eq (3.22) where $\mathbf{R}(t)^{-1}$ is performed within an iteration matrix inversion algorithm. They called it *Pseudo Recursive least squares* and they proved that the achieved solution is optimal for the model defined in Eq (3.12).

3.3.1 Least mean squares optical flow estimation

A strategy to avoid the inversion of the matrix and thus a framework with a huge computational cost is the introduction of an adaptive algorithm called *LMS* [4]. Interesting features of this method are: first, the simplicity implementation and second, the low computational cost. The algorithm is basically derived from an old searching algorithm: *the steepest descent*. To minimize the error in Eq (3.20) we used the gradient information of this error, thus after a number of iterations, we achieve asymptotically the minimum. The recursion is then described as:

$$\widehat{\mathbf{X}}(t) = \widehat{\mathbf{X}}(t-1) - \frac{\mu}{2}\frac{\partial\epsilon^2(t)}{\partial\mathbf{X}(t)} \quad (3.25)$$

Inside of this recursion the Correlation matrix and the Cross-correlation matrix introduced before, are present. Moreover, the term μ is the step size parameter that defines the weight of the searching method. The propagation of this recursion is computed over the time, frame by frame together with the updated equations of the variance and cross-correlation matrix in Eq (3.23). The recursion Eq (3.25) can be used iteratively within a computation that

considers just two following frames. This means computing the Covariance matrix and repeating the recursive Eq (3.25) many times. The results is the improvement of the accuracy and the performance. In fact, more iteration applied, more the result closes to minimum. The computation on the next frame deals with the previous estimation as the initial value and so continuously.

The Steepest descent method requires the propagation of the covariance and cross correlation matrix through the Eq (3.23). When introducing the LMS algorithm, this calculus is not needed anymore inside of the computation. This algorithm uses an instantaneous estimation of those matrices

$$\mathbf{R}(t) = \mathbf{H}^T\mathbf{H} + \beta\mathbf{S}^T\mathbf{S} \tag{3.26}$$

$$\mathbf{P}(t) = \mathbf{H}^T\mathbf{Y} \tag{3.27}$$

The recursion in Eq (3.25) can be then simply rewritten as:

$$\widehat{\mathbf{X}}(t) = \widehat{\mathbf{X}}(t-1) + \mu\mathbf{H}^T\mathbf{Y} - \mu\left(\mathbf{H}^T\mathbf{H} + \beta\mathbf{S}^T\mathbf{S}\right)\widehat{\mathbf{X}}(t-1) \tag{3.28}$$

Inside of this equation we notice that the forgetting factor is not present. The covariance matrix is not anymore propagated over time and the forgetting factor does not have any influence inside of the algorithm. For this reason the only suitable way to achieve the recursive propriety over time in the algorithm is to propagate the previous state to the next one through the initial condition.

Estimation scheme

In Fig (3.5) we sketch the estimation diagram of the least means square algorithm. We can see that the adaptation follows the normal loop of a LMS algorithm with the addition of a component due to the smoothness constraint. As we said before, the algorithm repeats the loop for a iterations number between two frames. The states calculated previously are simply propagated by setting the initial conditions on the next couple of images. It is possible to see how important the step size μ is, since it embodies the stability proprieties of the loop.

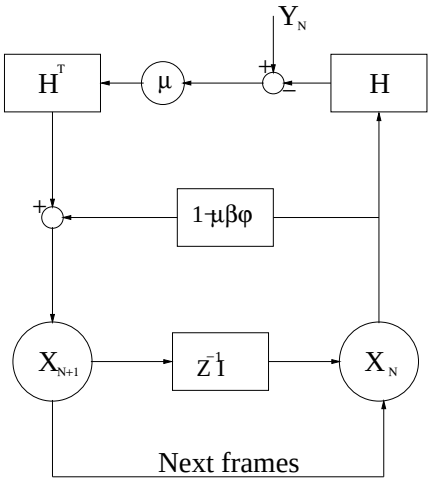


Figure 3.5: Optical Flow scheme estimation

3.3.2 Analysis of the algorithm and its implementation

The property of this algorithm comes from the value of the parameters inside of it: the step size parameter μ, the number of iteration, the variable β that weights the importance of the smoothness constraint. Another variable is the kernel associated to the smoothness constraint as we have seen in Eq (3.8). Both the composition and the size of this matrix influence the behavior of the algorithm. In fact, in the calculation of the field, it relates to the pixel where we compute the velocity with its neighbors. The bigger the matrix, the greater amount of pixels included in the computation. If this region belongs to the same moving object, the estimation operates more correctly. Instead, in the boundary, for instance between objects under movement and static background, it creates the effect of average between zero and not zero components. This, of course, contributes to a poor estimation. We do not study the performance for this particular parameters because it is more general problem of every optical flow algorithm, well documented in the literature, i.e [1].

The parameter β defines the weight of the smoothness constraint, associated to the matrix in Eq (3.8) and thus the problem of minimum in Eq (3.5). An increment of this parameter has the power to decrease the aperture problem, acting as an average filter in the neighborhood locations of a pixel. Increasing the parameter over a specific value can bring a bad estimation.

The iteration number allows the LMS algorithm to loop between two frames. Starting with the states previously calculated it performs the computation in order to bring to zero the mean square error. The larger the iterations number , the better result should be achieved. We inspect this parameter in the simulation part.

The most importance variable inside of a normal LMS algorithm is the step size μ. In fact from Fig (3.6) the presence of a feedback, inside of the algorithm, can involve some instability. The step parameter is a well known factor where a general LMS adaptation strategy can be designed [4]. The μ value is typically positive with an upper bound that avoids instability situation. The authors in [3] calculated an upper bound of this parameter and they formulated the following theorem:

Theorem 3.1 (Elad Feuer) *Consider the M-LMS algorithm as given in Eq (3.28) with arbitrary initial condition* $\mathbf{X}(0)$. *Let*

$$0 < \mu < \frac{2}{\lambda_{max}} \qquad where \qquad \lambda_{max} = \sup_{t>0} \left\{ \max_{1 \le k \le N_2} \{\lambda_k \{R(t)\}\} \right\} \tag{3.29}$$

Then $\exists\ 0 < \epsilon < 1$ *and* $0 < C_1 < \infty$ *such that*

$$\begin{aligned} \forall t > 0 \qquad \sigma_f(t) \;&=\; E\{|\mathbf{X}_m(t) - \mathbf{X}_{opt}(t)|\} \le [1-\epsilon]_{mt}\, \sigma_f(0) + \\ &+ \frac{\left[[1-\epsilon]_{m-1}\,(C_1+1) + C_1\right]}{1-[1-\epsilon]_m} \Delta_d \end{aligned} \tag{3.30}$$

This theorem respects the well known upper bound, typical for this kind of algorithm. In particular in the equation Eq (3.29) there is the explanation of choosing it through the selection of the maximum eigenvalue of the matrix $\mathbf{R}(t)$. Unfortunately the searching procedure is computationally not convenient because of the huge dimension of the covariance matrix. Instead, it is common to use an upper-bound of the maximum eigenvalues using the trace of the matrix as follows:

$$\lambda_{max} \le \sup_{t>0} tr(R(t)) \le \infty \tag{3.31}$$

The step size parameter is connected to the velocity of the adaptive estimation, as we can see from the theory described in [4]. With low values of

μ the adaptation is slow and the algorithm frame by frame has the *memory* effect. On the other hand, with higher values, the system reacts more rapidly to the changes but the mean square error tends to rise up. This means that we will have a degradation of the estimation performance.

In [3] the authors suggested searching method to help the choice of the step size parameter. In particular they use the method of *Normalized steepest descent (NSD)* to perform a line searching strategy inside of the loop between two frames. The recursive equation for μ is:

$$\mu_k = \frac{\mathbf{E}_k^T(t)\mathbf{E}_k(t)}{\mathbf{E}_k^T(t)\mathbf{R}(t)\mathbf{E}_k(t)} \tag{3.32}$$

where we have

$$\mathbf{E}_k(t) = \mathbf{P} - \mathbf{R}(t)\mathbf{X}_k(t-1) \tag{3.33}$$

We consider the same approach to have only a first look of the size of μ. In fact, to use this approach we need to propagate over the time the covariance and cross correlation matrices. The algorithm LMS does not need this propagation and thus it has the influence to speed the computation up.

3.4 Essential parameters estimation

Image motions on the camera plan can be structured with a eight-parametric model in Eq (3.4) that has been defined in chapter 2. We call the eight coefficients of this model the *essential parameters*. They are the target of this estimation part and we use them in the next chapter. We rewrite here the equations.

$$\begin{aligned} \dot{x} &= d_1 + d_3 x + d_4 y + \tfrac{1}{f}(d_7 x^2 + d_8 xy) \\ \dot{y} &= d_2 + d_5 x + d_6 y + \tfrac{1}{f}(d_7 xy + d_8 y^2) \end{aligned}$$

The dense computation of optical flow considers the velocity for each point of the image. To find the parameters d_i, in this case we apply a least square strategy, what results in an easy and immediate computation. The basic idea is to divide the image into regions and try to fit the correct model for every region. We described the variables in Eq (3.4) with the following notation.

$$\mathbf{X}(x,y) = \begin{bmatrix} 1 & x & y & x^2 & xy & 0 & 0 & 0 \\ 0 & 0 & 0 & xy & y^2 & 1 & x & y \end{bmatrix} \tag{3.34}$$

$$\mathbf{d} = \begin{bmatrix} d_1 & d_3 & d_4 & d_7 & d_8 & d_2 & d_5 & d_6 \end{bmatrix} \tag{3.35}$$

Two matrices above can be used as the regressors and the unknown variables in the least square computation. The essential parameters would be calculated in the region $\mathcal{R}$. Thus we formulate the estimation approach as:

$$\min_{d_{\mathcal{R}}} \sum_{x \in \mathcal{R}} \left| \mathbf{X}(x,y)\mathbf{d}_{\mathcal{R}} - \dot{\mathbf{X}} \right| \tag{3.36}$$

where we define $\dot{\mathbf{X}}$ as the optical flow estimation.

Sparse optical flow segmentation

The case with the sparse computation detected with a tracking algorithm functions somewhat easier. The valuation of the essential parameters can be performed with a limited number of points in order to obtain a well constraint least square estimation. This condition is satisfied if and only if the rank of the matrix in the Eq (3.36) is full. This means that the motion is uniquely determined if we perform the computation at four points on the plan and no three of them are collinear [2].

The sets of features that we collect on the images while tracking, have to be divided into groups in order to consider each group of points in planar surfaces. The approach is the same as it has been introduced for the dense computation. The estimated residual after the least squares operation has to be compared in respect with a given threshold to confirm that the group of considered points belongs to the same surface.

3.4.1 Reliability of the Optical Flow

The confidence measure of optical flow that cannot be forgotten to develop a efficient algorithm. The most dramatic problem of optical flow estimation derives from the incorrectness computation of some velocity vectors in the field, the outliers. In fact, aperture problem, boundaries and violation of the assumption that we have discussed before, concur to increase the problem of bad estimation.

Within a least squares scheme that evaluates the eight essential parameters, the presence of outliers is a common fact that has to be avoided in order to guaranteed the correct estimation. Least square is well known to be really sensitive to the outliers.

In [3] a possible solution for evaluating good estimation from the set of vectors is to inspect the diagonal of the correlation matrix R. The values smaller than a fixed threshold have to be discarded.

In this thesis we consider moving objects on a fixed background. For this reason we employ a simple strategy to inspect the value of the temporal gradient together with the norm of the velocity field. With the first method we inspect the part of the image subjected to motion. The second one focuses on the region where the moving objects produce a sensible optical flow estimation. In this way, we divide every image into two zones. One part is defined by the moving object and the second one is represented by the area where the optical flow results in a null value.

These two techniques of *segmentation* are not consider as fundamental for our work.

3.5 Simulations

In the current section we simulate the LMS algorithm presented before and we try to understand the behavior of the variable parameters inside of it. The estimated sparse computation has been included into the Kalman filter algorithm that will be explained in the next chapter.

The simulations of this algorithm were done inside of the virtual robot environment available at Lund University. The virtual scene was saved in memory and the further computations were done with the help of Matlab. To achieve better performance with this recursive algorithm we needed a long sequence of frame. For this reason we performed the simulation up to 80 frames. Moreover we added a noise component to the virtual scene. We chose Gaussian noise with mean zero and variance $\sigma = 5$. This noise model was added over the 225 gray levels of each frame.

The image processing part was carried out with the aid of Matlab. In particular the spatiotemporal gradients of the image were performed through the matlab function "gradient". This operation is similar to the one proposed by Horn and used in [3].The case is to convolve the kernel $\begin{bmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix}$ over all the three dimensions of the image stream. Some critics have been

pointed out to this kernel for the not optimal characteristics that it performs We did not try other kernels or methods and we leave this for future work. Before starting the computation of the estimation we decided to smooth the image with a blurring filter with a Gaussian kernel 5x5. It is well known that this operation rises down the effect of the noise and then improves the estimation process. The performance of the algorithm was studied in the qualitative method. The size of the pixels matrix and the number of variable inside of it do not help to find a good choice to evaluate the behavior of the algorithm. Another problem that appears is that for different kinds of patterns the algorithm can give different sorts of results. In [3] the simulations are performed with a semi synthetic video sequence. The comparison of the algorithms inside of it has been done through the mean square error between the estimated flow and the real one.

3.5.1 Qualitative analysis

We show in Fig (3.6) the result of the algorithm performance with three cubes. We chose fairly simple examples of motion to perceive better behavior of the algorithm. In fact, one cube is rotating on Z-axis, one on Y axis and the last one is translating. Moreover we consider some cases with different patterns. In patterns such as chess textures, an optical flow algorithm has more possibilities to achieve a good result, since the aperture problem is decreased to zero.

In the next set of figures, Fig (3.7), we show the behavior of the algorithm over the time and the changes that the field is subjected to. For simplicity we moved the cube with a simple rotation on the Z axis. A complicate and non regular pattern was drawn in the surface in order to evaluate the aperture problem.

Finally we tested a case where we consider a real sequence of image to test the algorithm. In Fig (3.8) we have a Rubik cube that is fixed on a rotating platform. We succeeded with correct estimation for the platform but there are some inaccuracies for the cube, which are related to the aperture problem. We apply the least mean squares in a sequence of 20 images and seems that the algorithm needs more. We show in Fig (3.9)a particular piece of image to evaluate better and to present more clearly the estimation result.

In general the algorithm has worked fairly well, but is not completely satisfactory. The field on the boundary in most cases suffers from aperture problem, which was not eliminated in every site.

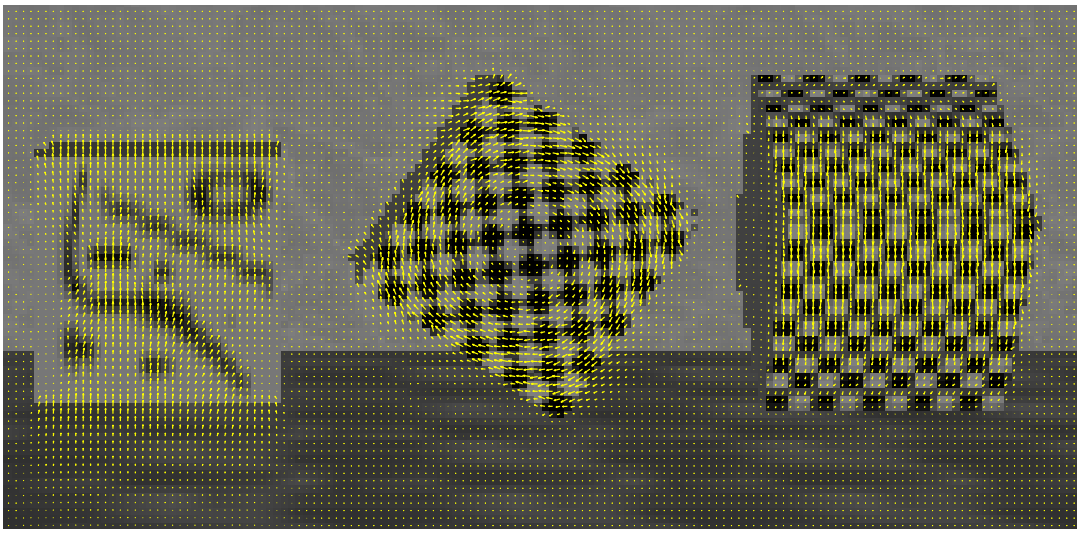
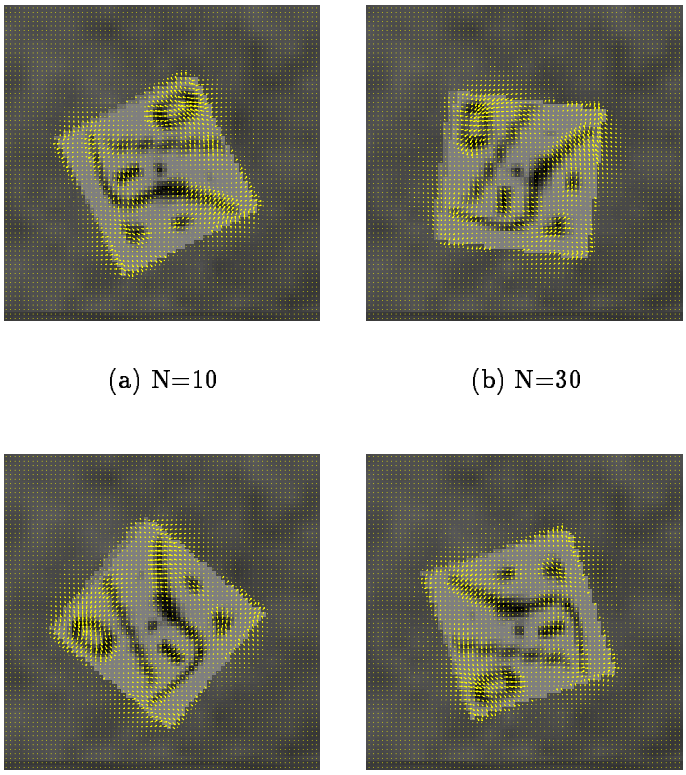


Figure 3.6: LMS algorithm

3.5.2 Parameter analysis

The second analysis that we carried out regarded the parameters inside the LMS algorithm. We inspected the behavior of μ, the number of iterations and the variable β. The approach that the used consists of collected sequence of images from cube subjects to a vertical constant motion. We drew a simple chess pattern on it. The following analysis was performed when changing every time one parameter and keeping constant the other two. To evaluate those characteristics appropriately, we consider only one section of the image, in order to plot the vertical velocity in a graph. In Fig (3.10) the particular image section under analysis has been marked in red.

In the first experiment we investigated the μ, with β and the iterations constants. In the first stage we went through the algorithm using the NSD line searching approach. In this way we can realize the magnitude order of μ. In Fig (3.11(a)), after running the algorithm for the cube, we have the evolution of μ. It is possible to see that after a transient period, the step size parameter achieves a constant value. This value gives us a starting point in order to change it in our analysis. We calculated also the bound for the step size, described by the Th (3.3.2). We get $\mu = 2/\lambda_{max} = 1.07e^{-8}$. This value is too distinct from the μ calculated previously and does not give us a good



(a) N=10 (b) N=30

(c) N=50 (d) N=70

Figure 3.7: Evolution of a rotating cube!

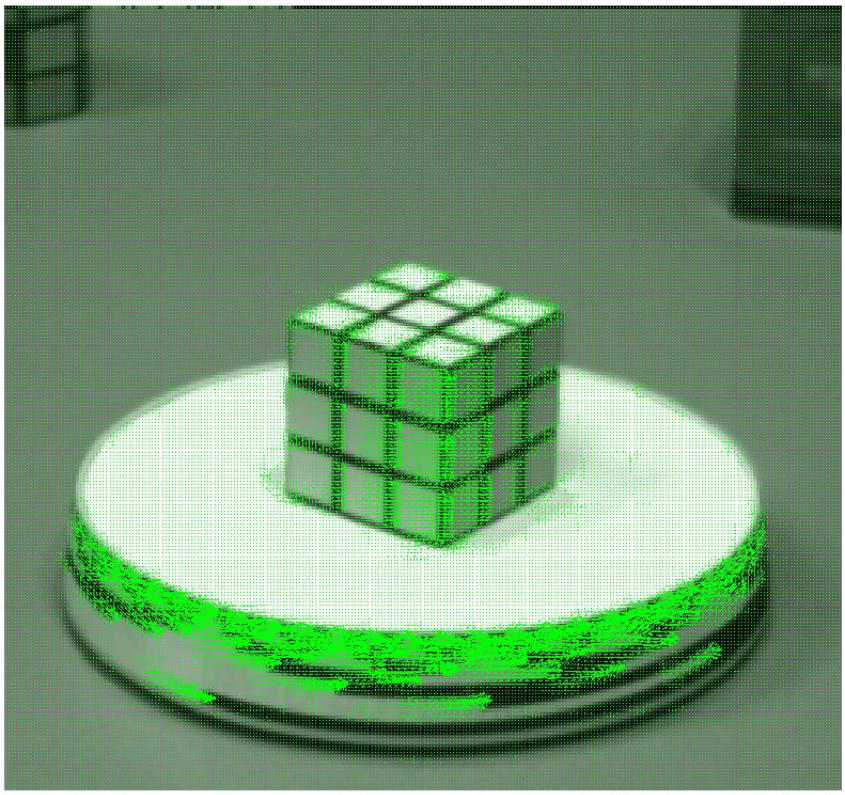


Figure 3.8: Rubik cube

precision.

In Fig (3.11(b)), we can see the effect of our analysis. The algorithm results in a better performance when increasing the parameter μ. This result can be considered as reasonable. In fact, we can see that the higher the step size , the faster the system reaches the real value. Small values of μ contribute to the increase of *memory* effect that in the case of translating surfaces induces a *tail* effect in the flow.

The next stage considers the parameter β and we plot the results in

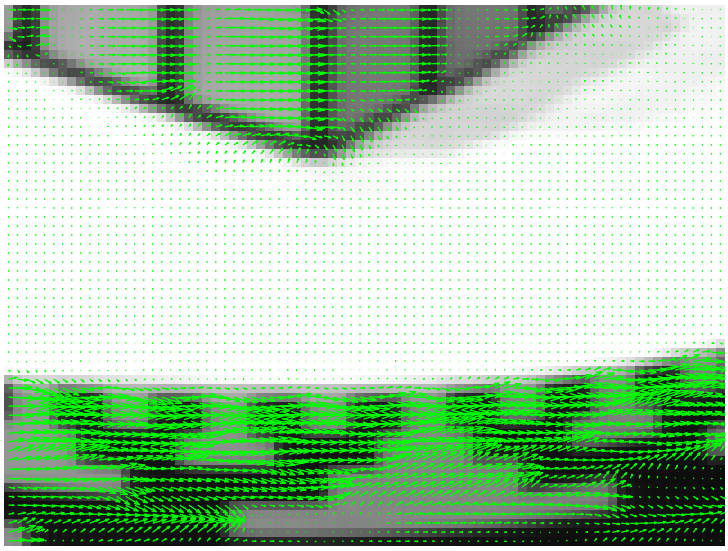


Figure 3.9: Rubik cube

Fig (3.12(a)). We can see that an increment of this parameter has the effect to decrease the value of the optical flow intensity. The situation seems generated from the fact that the smoothness constraint calculates the average between different pixel locations. In fact, the velocity field for the chess pattern should be less intensive where the brightness pattern is constant and high where there is the step between two adjacent squares. Increasing the level of β rises the average between these two values. If the parameter is set up too low, it contributes to system instability, as we can see from the case $\beta = 1$ in the graph.

At the end we inspected the number of iteration for the LMS algorithm shown in Fig (3.12(b)). The intensity level does not change in the locations where the cube is passing. Instead the "memory" effect can be raisen down with an increment of the iteration number. With a number of 10 iterations we can achieve a good result without increasing too much the computation time.

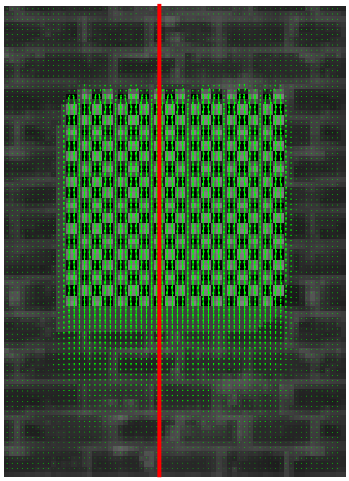
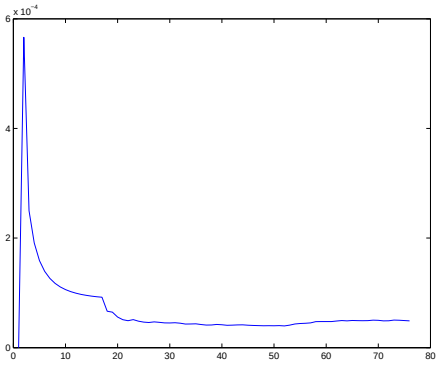


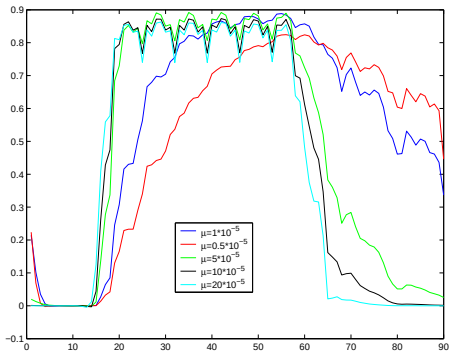
Figure 3.10: Translating cube

3.5.3 Segmentation

The last section comprehends the estimation of the essential parameters. We use the least square algorithm that has been proposed before.In order to achieve the computation of the eight parameters we have to distinguish the regions where the computation has to be done. In Fig (3.13) the case of a rotating planar surfaces has been taken under consideration. After evaluating the optical flow we find the zone where the plane produce a remarkable optical flow estimation and in that zone we consider the least squares estimation scheme.

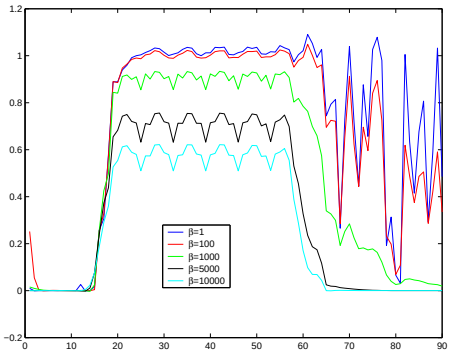


(a) Value of μ using the NSD line search approach

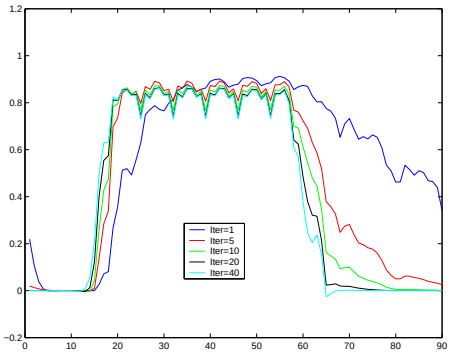


(b) Effect of changing the step size with $\beta = 2000$ and iterations number 10

Figure 3.11: Analysis for the step size parameter μ



(a) The parameter β with $\mu = 5 * 10^{-5}$ and the number of iteration constant at 10



(b) Effect of changing the Iteration number with $\beta = 2000$ and $\mu = 5 * 10^{-5}$

Figure 3.12: Analysis for the parameters of LMS

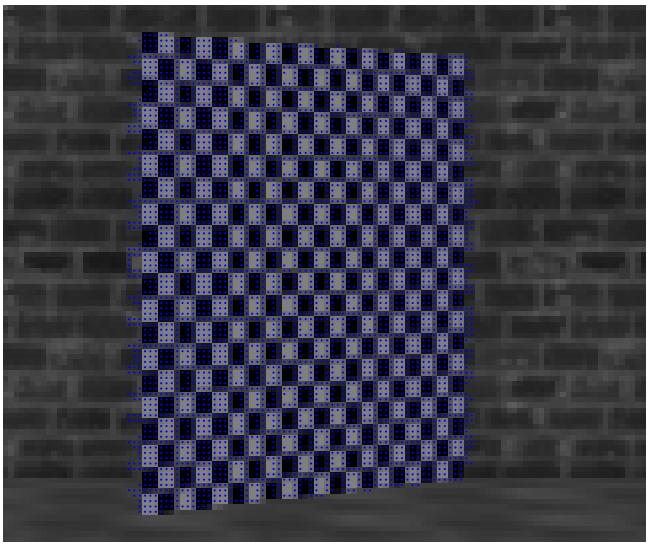


Figure 3.13: Segmentation of the moving object

Chapter 4

Dynamic Vision

The problem of estimating the motion and structure parameters is a typical problem inside of the field of computer vision. In the last two decades it has become an important subject of investigation for many interesting applications. We conclude with this chapter the framework for recovery shape from motion. We consider a first part where we study a new class of perspective system theory. The last part we test two algorithms in order to achieve the shape parameters.

4.1 Perspective system theory

The formulations and the solutions of the problems that have arisen in computer vision system are strongly dependent on a camera model that observes either a moving or static scene. As we have seen in chapter 2 the most typical and simple description of a camera is the pinhole model. The main characteristic of this kind of camera is that it observes a point in the space up to homogeneous line. This consideration arises the introduction of a new class of problems in the system theory, which designs a valid model to deal with appearing obstacles and includes, for our purpose, some dynamics evolutions. The new subject has been called *Perspective System Theory* and has been investigated for the last ten years within a long list of papers, i.e. [5][8][6] [7]. It has a great influence on the motion estimation and we find it interesting to study for a possible utilization where it can be applied. A perspective system is a linear or nonlinear dynamic system whose output is observed perspectively. We describe this system introducing the following

formulation:

$$\begin{cases} \dot{x}(t) = f(x(t)), \quad x(0) = x_0 \in \mathbf{R}^n \\ y(t) = h(Cx(t)) \end{cases} \tag{4.1}$$

where we have the state $x(t) \in \mathbf{R}^n$ and the rational function $f : \mathbf{R}^n \to \mathbf{R}^n$. The $y(t)$ is the *perspective observation* of the state $x(t)$, where we define the perspective observation function as:

$$\begin{aligned} &Y : \mathbf{R}^n - B \to \mathbf{RP}^{m-1} \\ &x \mapsto [Cx] \end{aligned} \tag{4.2}$$

where $B = \{x \in \mathbf{R}^n : Cx = 0\}$ and where $[Cx]$ is the vector of the homogeneous coordinates of Cx as an element of $\mathbf{RP}^{m-1}$, the m-1 dimensional projective space of all homogeneous lines in $\mathbf{R}^m$.

The definition in eq (4.1) gives a general description of the class of the perspective system. In particular the state dynamic is controlled by f, a rational function. The dynamics, instead, is typically described by a simple matrix notation in the following form:

$$\dot{x} = \mathbf{A}x + \mathbf{b} \tag{4.3}$$

4.1.1 Simple perspective system

A simple example of the perspective system is carried out directly from the case of a 3D point subjected to a rigid transformation. In this case the dynamics is constituted by the Eq (4.3) where the rotation matrix A is skew-symmetric. The observation function that we analyze from a pinhole camera, describes the point from a 3D space in a 2D moving feature. The considered system is the one denoted in Eq (4.4):

$$\begin{cases} \dot{x} = \mathbf{A}x + \mathbf{b}, \quad x(0) = x_0 \in \mathbf{R}^n \\ y(t) = [x(1)/x(3)x(2)/x(3)] \end{cases} \tag{4.4}$$

The formulation of the observation function is define as:

$$\begin{aligned} &Y : \mathbf{R}^3 - B \to \mathbf{RP}^2 \\ &(x, y, z) \mapsto [x, y, z] \end{aligned} \tag{4.5}$$

where the expression $[x, y, z]$ defines the vector observed perspectively on the image plan $(x/z, y/z)$, the second dimensional projective space.

4.1.2 Problems in perspective system theory

From the perspective system, that we have just defined, arise two particular problems that have been analyzed in the literature and represent the kernel of the studying of this theory. We define as follows the formulation of these problems taken, i.e. from [5]

Problems 4.1 ***[Perspective observability problem]*** *Consider the dynamic system in Eq (4.3) together with the observation function given by Eq (4.2). Moreover, assume that the parameters in A,b are known and assume that one observes the output* $[y(t)]$ *up to a homogeneous line. The problem is to estimate* (x_0, y_0, z_0) *from the observed data.*

Problems 4.2 ***[Perspective Identification problem]*** *Consider the dynamical system in Eq (4.3) together with the observation function given by Eq (4.2). Moreover, assume that the parameters in A,b are unknown and assume that one observes the output* $[y(t)]$ *up to a homogeneous line. The problem is to estimate A,b and the initial condition* (x_0, y_0, z_0) *from the observed data.*

Both problems have a big influence inside of the application that requires the use of camera as, for instance, robotics. The first one include information about the dynamics of the system and the requirement is the only estimation of the state. The main problem is to detect some conditions of observability of the states in order to be able to build a valid observer. If we think to a mobile robot in an unknown ambient with a camera encapsulated on it, the dynamics can be evaluated by some sensor from fixed points in the 3D space. The scene observed by a camera, undergoes by ego-motion to dynamic transformation. The question is if we can estimate geometry of the objects, position of the robot and which are the condition that needs to fulfill. In [7], the authors check the observability condition through an extension of the famous *Hautus* test.

Identification problem

The second problem is more complex in face of complete absence of knowledge about the system. The solution excels in finding a system that realizes the perspective output together with the exact structure $\mathbf{A}$,$\mathbf{C}$ and recovers the initial state $\mathbf{x_0}$. This is not an easy task for the elevated number of choices.

The following theorem from [5] delineates the possible solution up to an orbit of a group action.

Theorem 4.1 (Ghosh&Loucks[5]) *Consider the perspective system in continuous time*

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{4.6}$$
$$\mathbf{y} = [\mathbf{C}\mathbf{x}] \tag{4.7}$$

where we assume that the triplet $(\mathbf{A}, \mathbf{C}, \mathbf{x_0})$ *is minimal. The set of all minimal triplets which produces the same output* $\mathbf{y}$ *is given precisely by the orbit of a group* $\mathcal{G}$ *whose action is described by:*

1. $\mathbf{P} \in GL(n)$ *acting on* $(\mathbf{A}, \mathbf{C}, \mathbf{x_0})$ *as follows*

$$(\mathbf{A}, \mathbf{C}, \mathbf{x_0}) \mapsto \left(\mathbf{P}^{-1}\mathbf{A}\mathbf{P}, \mathbf{C}\mathbf{P}, \mathbf{P}^{-1}\mathbf{x_0}\right) \tag{4.8}$$

2. $\lambda_1, \lambda_3 \in \mathcal{R} - 0$ *acting on* $(\mathbf{A}, \mathbf{C}, \mathbf{x_0})$ *as follows*

$$(\mathbf{A}, \mathbf{C}, \mathbf{x_0}) \mapsto (\mathbf{A}, \lambda_1\mathbf{C}, \lambda_3\mathbf{x_0}) \tag{4.9}$$

3. $\lambda_2 \in \mathcal{R}$ *acting on* $(\mathbf{A}, \mathbf{C}, \mathbf{x_0})$ *as follows*

$$(\mathbf{A}, \mathbf{C}, \mathbf{x_0}) \mapsto (\lambda_2\mathbf{I} + \mathbf{A}, \mathbf{C}, \mathbf{x_0}) \tag{4.10}$$

The former theorem restricts the choices of realization to an orbit of a group that has been called *perspective group*. The first action on this group in Eq (4.8) is well known. It describes the operation of changing basis in the state space. The other two Eq (4.9) Eq (4.10), are the consequence of the perspective observation function.

The same conclusion can be found for a discrete time system. In [6], the authors calculate the action of the particular group $\mathcal{G}$ for this specific case.

The conclusion states that it is impossible to recover completely the parameters $(\mathbf{A}, \mathbf{C}, \mathbf{x_0})$ from a general perspective system. We need to evaluate case by case the structure of the system to find positive or negative answer.

4.2 Shape from motion

As it has been described, for the problem of shape from motion we defined a particular algorithm that implements the estimation of the dynamics and the geometrical structure of the scene for images produced by a camera in various time steps. In chapter 2 we define in details the dynamics of moving planar surfaces. In particular we have observed that if the plan undergoes a rigid body transformation, the shape parameters (p, q, r) follow a Riccati differential equation called Shape dynamics. From this analysis, we would like to build a state space system that for the intrinsic propriety of the camera has the formulation inside of the perspective system theory.

We start with considering the shape dynamics derived from the rigid body motion applied to the planar surface. We rewrite the equation that has been calculated in chapter 2:

$$\frac{d}{dt}\left(\bar{p}, \bar{q}, -\bar{s}, \bar{r}\right)^T = \mathcal{A}\left(\bar{p}, \bar{q}, -\bar{s}, \bar{r}\right)^T \tag{4.11}$$

where

$$-\mathcal{A}^T = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 0 \end{pmatrix} \tag{4.12}$$

We would like to set up a system that considers the identification of the shape parameters choosing an appropriate output function. Since we have inspected the motion of the surface through the optical flow estimation described in chapter 3, we can assume to have the 8 essential parameters of the moving plan. Unfortunately, as we can see from the Prep (2.1) the d variables present nonlinearity on the shape and dynamic parameters. It is necessary to homogenize the vector of essential parameters as follows:

$$d_i = \frac{y_i}{y_9}, \qquad i = 1, \ldots, 8 \tag{4.13}$$

With this particular operation we consider the new vector $(y_i, \ldots, y_9)$ having the output of the matrix multiplication given by:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -fb_1 & fa_{13} \\ 0 & 0 & -fb_2 & fa_{23} \\ -b'_1 & 0 & b_3 - \delta a_{11} & a_{11} - a_{33} \\ 0 & -b'_1 & -\delta a_{12} & a_{12} \\ -b'_2 & 0 & -\delta a_{21} & a_{21} \\ 0 & -b'_2 & b_3 - \delta a_{22} & a_{22} - a_{33} \\ -b'_3 & 0 & -\delta a_{31} & a_{31} \\ 0 & -b'_3 & -\delta a_{32} & a_{32} \\ 0 & 0 & -\delta & 1 \end{pmatrix} \begin{pmatrix} \bar{p} \\ \bar{q} \\ -\bar{s} \\ \bar{r} \end{pmatrix} \tag{4.14}$$

where we have used a new vector $\mathbf{b}'$ that has been calculated by:

$$(b'_1, b'_2, b'_3) = (b_1 - a_{13}\delta, b_2 - a_{23}\delta, b_3 - a_{33}\delta) \tag{4.15}$$

The parameter δ, as described in chapter 2, is the parameter that has been introduced to employ the possibility of choice between the image centered and the viewer centered coordinate system. We assign the symbol Δ to the matrix of the observation function.

Finally we can deduce from the former investigation that the combination of Eq (4.11) and Eq (4.14)that both inspect the problem of shape from motion can be referred as an example of perspective system.

The dynamics of this system can be easily extended to the particular case of the rigid body transformation. This can be done including a skew-symmetric matrix in Eq (4.14).

4.2.1 Identifiability and realization of the system

The main question of the shape from motion is to reconstruct the structure of the scene by the evolution of the 3D world. This specific case is included in the definition of the problem 2 above, where it claims to identify the whole state space system by the output observed in a specified time interval $[0, T]$. We would like to ask if it is possible and what are the condition to estimate the following parameters:

$$A, b_1, b_2, b_3, p, q, r \tag{4.16}$$

together with the initial condition $, p(0), q(0), r(0)$. For the simple case of rigid body, there are 9 parameters and then we have:

$$\omega_1, \omega_2, \omega_3, b_1, b_2, b_3, p, q, r \tag{4.17}$$

The identification of these parameters is concentrated in finding the particular realization of the system that performs the same output with exactly the same dynamics. As we have seen in the introduction of the perspective system the sets of choices for the matrices $\mathcal{A}$, $\mathbf{\Delta}$ and p_0, q_0, r_0 that produce the same output $d_{1..9}$ are in the orbit of the group $\mathcal{G}$ described in Eq (4.8)Eq (4.9)Eq (4.10). Under generic condition the identification approach is impossible.

Instead, from the particular structure of the system in Eq (4.11) and Eq (4.14), we can detect some particulars in order to restrict the number of choices and then permit the identifiability. The main results are due to Ghosh&Loucks over the perspective system theory and are proved in two papers, [5] and [6], for the continuous and for the discrete time cases, respectively. The following theorem explains the result:

Theorem 4.2 (Ghosh&Loucks) *Consider the perspective system Eq (4.11) Eq (4.14) and assume that the triplet $\mathcal{A}, \mathbf{\Delta}, \mathcal{P}$ is minimal. Moreover, assume that the generic conditions hold:*

$$b'_1 a_{23} - b'_2 a_{13} \neq 0 \quad b'_2 a_{31} - b'_3 a_{21} \neq 0 \quad b'_1 a_{32} - b'_3 a_{12} \neq 0 \tag{4.18}$$

$$(b'_1, b'_2, b'_3) \neq 0 \tag{4.19}$$

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{b}'^T \end{pmatrix} \quad \textit{has rank 3} \tag{4.20}$$

where $\mathbf{b}'^T = (b'_1, b'_2, b'_3)$ *and*

$$b'_1 = b_1 - a_{13}\delta \tag{4.21}$$

$$b'_2 = b_1 - a_{23}\delta \tag{4.22}$$

$$b'_3 = b_3 - a_{33}\delta \tag{4.23}$$

Under these conditions the parameters that can be locally identified via perspective observation are:

$$(\mathbf{A}, p, q, c_1, c_2, c_3) \tag{4.24}$$

where c_1, c_2, c_3 are the relative translation velocities defined previously define by

$$c_i = \frac{b_i - a_{i3}\delta}{r + \delta} \tag{4.25}$$

The condition of identifiability is due to the special structure of the matrices on the perspective system.

The identifiability theorem gives us the information, under the described condition, that there is the possibility to recover the matrices and the states during the observation. The next stage is to find an estimator in order to achieve this goal. In the first part we consider a classical approach that deals with the solution in an analytical way. In the second we would like to study a recursive and adaptive method that for real and noisy application has more suitable results.

4.2.2 An Analytical approach

In this section we study a particular algebraic approach in the way to resolve the problem of shape from motion estimation with moving planar surfaces. The analytical method is the most used in the literature and in this thesis we follow the interesting algorithm proposed by Kanatani in [2] and suggested in [5]. It is an interesting application of the theory of geometrical invariance and group representation based on the *Weyl's thesis*.

As we have discussed in the introduction, the purpose of the estimation is to achieve a set of scene structures through the corresponding characteristics projected onto the image plan. The relationship between them are described most of the time by nonlinear functions, as in Eq (4.26), arising the problem to compute the scene parameters $\alpha_1 ... \alpha_n$. The equations in Eq (4.26) have been called *3D recovery equations* and in [2] the goal is to find an analytical solution of them.

$$\beta_j = F_j(\alpha_1 ... \alpha_n) \qquad j = 1, ..., m \tag{4.26}$$

Invariants of Optical Flow

The kernel of the approach of Kanatani consists in the inspection of the motion due to a rigid body transformation in terms of invariance proprieties.

The idea is to identify some parameters that keep constant their characteristics over the motion, in order to associate to them the scene structures. These invariants have particular own meaning, separate from each others in a geometrical and a dynamical point of view. In [2] has been explained the general details of this invariance theory and we refer that text for more information.

For our particular purpose , we are interested to associate to the eight essential parameters of a moving planar surfaces, some specific structure proprieties. In this case in [2] we have the following invariants from the parametric model in Prep (2.1) due to rigid body motion:

$$
\begin{aligned}
U_0 &= d_1 + id_2 && (4.27)\\
T &= d_3 + d_6 && (4.28)\\
R &= d_4 + d_5 && (4.29)\\
S &= (d_3 - d_6) + i(d_4 + d_5) && (4.30)\\
K &= d_7 + d_8 && (4.31)
\end{aligned}
$$

where, U_0 represents the translation velocity, T the divergence, R the rotational velocity, S the sharing and K the fanning of the plane that undergoing the transformation.

Together with the invariants that we have just mentioned there are others derived from the shape parameters. They do not have a particular meaning but they are important for the construction of the 3D recovery equations:

$$P = p + iq \qquad W = \omega_3 - i\omega_2 \qquad V = c_1 + ic_2 \tag{4.32}$$

Solution of 3D recovery equation

The invariants that we have collected from the plane model can be used to formulate an algebraic approach for shape for motion estimation. The Weil's theorem considers the possibility to construct equations adding terms belong to the same class of *weight*. The application of this theorem, in the case of shape from motion, brings to the formulation of the 3D recovery equations that we have cited above.

Without entering in the particular construction of these equations we consider the following important theorem formalized by Kanatani and cited

by Ghosh in some his papers as an interesting solution of the scene Reconstruction problem.

Theorem 4.3 *Assume that $c_3 \neq 0$ then the system described by Prep (2.1) can be solved for exactly two real solutions. If*

$$(c_1, c_2, c_3, \omega_1, \omega_2, \omega_3, p, q) \tag{4.33}$$

is one solution, then the other solution is given by

$$(-c_3 p, -c_3 q, c_3, \omega_1 - c_1 q + c_2 p, \omega_2 + c_1 + c_3 p, \omega_3 + c_2 + c_3 q, -c1/c3, -c2/c3) \tag{4.34}$$

Subject to the condition of the relative translation velocity c_3 is different from zero, the solution of the 3D recovery equation is ambiguous. The author in [2] declares the visual ambiguity is well known between psychologists but we humans do not suffer it for the fact that we employ more complex mechanism of recognition.

The two solutions in Th (4.3) can be solved inspecting directly the 3D recovery equations. The computation that is explain in [5] follows the next algorithm:

$$c_3^3 + T c_3^2 + \frac{1}{4}\left(T^2 - |L|^2 - |S|^2\right) c_3 + \frac{1}{8}\left(\Re\left(L^2 S^*\right) - T\,|L|^2\right) = 0 \tag{4.35}$$

$$V = \frac{-L \pm \sqrt{L^2 - 4c_3 S}}{2} \tag{4.36}$$

$$P = \frac{L \pm \sqrt{L^2 - 4c_3 S}}{2c_3} \tag{4.37}$$

$$W = i\left(V - \frac{1}{f} U_0\right) \tag{4.38}$$

$$\omega_1 = \left(\Im(PV^* - R)\right)/2 \tag{4.39}$$

$$\tag{4.40}$$

where $L = fK - \frac{1}{2}U_0$.

In order to achieve the two solutions defined by the Th (4.3), we need to consider in Eq (4.35) only the middle root of c_3.

Resolve the ambiguity

There is not a way to resolve the ambiguity with the information that we have for a planar surfaces. In [2] the only possible approach is to look to other patches belonging to the same object. In this case, in fact, it is possible to descend which solutions is the correct one inspecting the rotational velocity. These parameter do not change during the computation of the optical flow even if we change reference. In chapter 3 we have seen that the eight essential parameters have been calculated using a least squares estimation. We need to perform such operation respect to a reference point which is the center of the rotation. If we choose a different reference point, as described in [2], we have the same rotational velocities but different in translation.

4.2.3 Extended Kalman filtering approach

The analytical solution described in details in the previous section suffered from a noise problem that is rather typical for this kind of method. Instead, in an engineering application, it is preferable to use a recursive approach that integrates the observed data with the adaptive estimation in order to cope the noise problem. The use of the Kalman filter, for a big class of typical noise condition is an optimal method. Moreover, as we have declared in the previous chapters, Kalman filtering has also long history for many computer vision applications.

The shape from motion issue for the case of moving planar surfaces, a we have seen, is described by a set of nonlinear equation inside of the dynamics equations. For this reason, we consider an estimation algorithm that consists on the use of an *Extended Kalman filter*. We support our work with the approach proposed in [8], where the realization of the filter has been studied.

Moreover, inside of the mathematical formulation of the filter, the algorithm of tracking features is included. In this condition we only need to employ the position of the pixels location under tracking.

Filter equation

The formulation of the recursion is based on the study of the optical flow dynamics described in chapter 2 for the case of discrete system. Under this optic we designed the filter considering the following state vector θ:

$$\theta_k = (a_{11,k}, \ldots, a_{33,k}, c_{1,k}, c_{2,k}, c_{3,k}, p_k, q_k, x_{1,k}, y_{1,k}, \ldots, x_{n,k}, y_{n,k})^T \quad (4.41)$$

where the first 9 coefficients $a_{ij,k}$ are the elements inside of the discrete rotation matrix, the $c_{i,k}$ are the relative velocity translation, the p and q the shape parameters and at end a set of n points collected from the image features. We refer the understanding of these symbols to the discrete analysis of the planar motion. The state dynamics equation is a nonlinear function in the previous parameters. The state space system for the EKF is given by:

$$\begin{aligned} \theta_{k+1} &= f(\theta_k) \\ z_k &= H\theta_k + w_k \end{aligned} \quad (4.42)$$

where the nonlinear space dynamics function is a collection of the following equations:

$$a_{ij,k+1} = a_{ij,k} \qquad i,j = 1,2,3 \quad (4.43)$$

$$c_{i,k+1} = \frac{(\Delta_7 x_k + \Delta_8 y_k + \Delta_9)\, c_{i,k}}{\bar{c_k}} \qquad i = 1,2,3 \quad (4.44)$$

$$p_{k+1} = \frac{\Delta_1 x_k + \Delta_2 y_k + \Delta_3}{\Delta_7 x_k + \Delta_8 y_k + \Delta_9} \quad (4.45)$$

$$q_{k+1} = \frac{\Delta_4 x_k + \Delta_5 y_k + \Delta_6}{\Delta_7 x_k + \Delta_8 y_k + \Delta_9} \quad (4.46)$$

$$x_{i,k+1} = \frac{d_1 x_k + d_2 y_k + d_3}{d_7 x_k + d_8 y_k + d_9} \quad (4.47)$$

$$y_{i,k+1} = \frac{d_4 x_k + d_5 y_k + d_6}{d_7 x_k + d_8 y_k + d_9} \qquad i = 1 \ldots n \quad (4.48)$$

where

$$\bar{c_k} = (\Delta_1 x_k + \Delta_2 y_k + \Delta_3) c_{1,k} + (\Delta_4 x_k + \Delta_5 y_k + \Delta_6) c_{2,k} + \\ + (\Delta_7 x_k + \Delta_8 y_k + \Delta_9) c_{3,k} - 1 \quad (4.49)$$

and the parameters $d_{i,k}$ are defined in Eq (2.24).

The output function is defined through the observation of the moving features on the image plane. Thus we can build the output matrix, simply as

$H = [0_{2n,14} I_{2n}]$. The dimensions fit the state that is composed by $\theta_k \in \mathbf{R}^{2n+14}$ and the output by $z_k \in \mathbf{R}^{2n}$. We consider, of course, a contribution to a Gaussian noise $w_k \in \mathbf{N}(0, R)$ with 0 mean and R covariance.

Concluding the analysis, we introduce the extended Kalman filter through the well known algorithm. The filter equation is given by:

$$\begin{aligned}
\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + \mathbf{K}_k \left[z_k - \mathbf{H}\hat{\theta}_{k|k-1} \right], \qquad \hat{\theta}_{0|-1} = \bar{\theta}_0 && (4.50)\\
\hat{\theta}_{k+1|k} &= f\left(\hat{\theta}_{k|k}\right) && (4.51)\\
\mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}^T \left[\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R} \right]^{-1} && (4.52)\\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k|k-1}, \qquad \mathbf{P}_{0|-1} = \Sigma_0 && (4.53)\\
\mathbf{P}_{k+1|k} &= \mathbf{F}_k\mathbf{P}_{k|k}\mathbf{F}_k^T && (4.54)
\end{aligned}$$

where we consider the Jacobian matrix $\mathbf{F}_k$ as follows:

$$\mathbf{F}_k = \left. \frac{\partial f(\theta)}{\partial \theta_T} \right|_{\theta=\hat{\theta}_{k|k}} \tag{4.55}$$

In [8] the authors consider the introduction of the EKF a discussion on the use of a recursive estimation scheme for the shape from motion problem. In the article has been pointed out that under particular conditions the shape parameters are *hidden* and the computation performs an incorrect result. If $\mathbf{b} \in Im(\omega)$ then the recursive approach results in a valid solutions. Otherwise the shape parameters are invisible from the image data and the estimation would not be possible.We do not enter in the details of this special condition and we do not use the results of these studies inside of the thesis but it is important to mention it.

4.3 Simulations and results

The simulation guideline that we have followed, uses the synthetic 3D world and in the next stage the Virtual Environment. This part is organized to consider the case of analytical solution proposed by Kanatani and the extended Kalman Filter. The screen shots relative to these two 3D worlds are in the Fig (1.2) and Fig (1.3) in the introduction.

4.3.1 Analytical solution

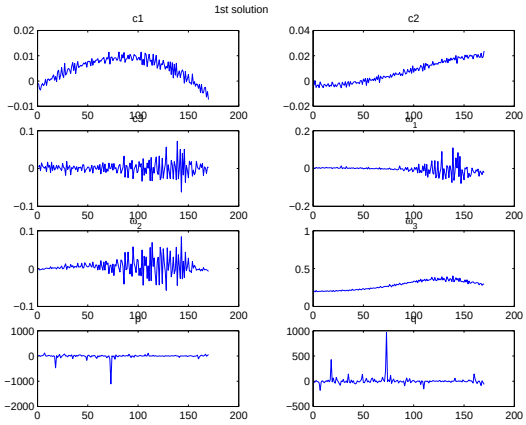
In this section we consider some simple example for the algorithm suggested by Kanatani. We have carried out our analysis in the 3D world construct in Matlab. In the first figure we show the experiment with one planar surface. The camera was placed on the zero of the three Cartesian axes and the two plans to a distance of 1 unit. The rigid transformation that we apply to the plane, allows the camera to see correctly the surface during the whole motion. We chose a reference point for the rotation at $P_0 = (0, 0, 3)$, on the camera axis. The velocity of the equivalent rigid body was set up as follows: $(\omega_1, \omega_2, \omega_3) = (-0.2, 0.1, -0.1)$ and the translation velocity equal to $(b_1, b_2, b_3) = (0.1, 0.1, -0.1)$. With these values the plane keeps his orbit close to the camera and allows the parameters p and q to assume limited values. Moreover, the condition for a correct recursive estimation, in case of the use of the EKF, is fulfilled.

In Fig (4.1), we have analyzed the motion by two solutions. We chose a Gaussian noise with level 0.005 unit. Even if we do not look at different surfaces, we can see from the shape coefficients that the first set of solution is completely missed. In fact, the coefficient p is strongly not stationary and it is changing between high and low values. The strategy of observing the parameters evolution can be included to discard the ambiguity.

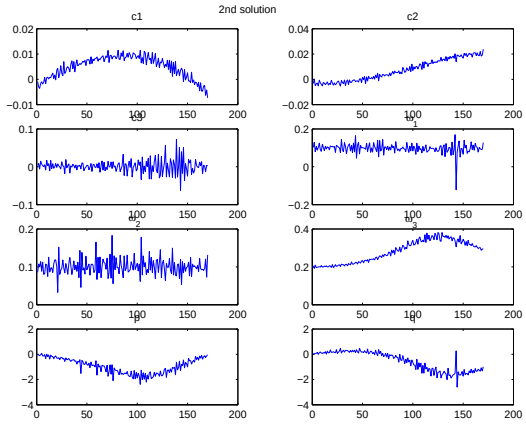
In the second set of figures we include the possibility to rise the level of noise. In particular we show the velocity rotation ω_3 around the optical axis. The noise level is the same as in the next case of EKF in order to compare the results. In Fig (4.2), we can check the strong influence of noise with this algorithm. The last figure is showing that the effect of noise can *hide* the motion parameters and thus the estimation is incorrect.

In the third part of simulation we focus on the possibility to observe two different planar surfaces but connected to each other by the same rigid body motion. We rotate the two connected planar surfaces with the same velocity $\omega_2 = 0.1$. We show the planar surfaces in Fig (4.3). In Fig (4.4) between four solutions we present one that is considered as correct. From these figures the velocity ω_3 contains more or less the same values. If we have a look of the shape parameter p we can argue that the the planar surfaces has the same evolution but starting from a different point.

The final stage of the 3D recovery equation algorithm investigation is effected in the Virtual Robot Environment. We consider the same situation as described in chapter 3 regarding the pre-image-processing stage. In particu-



(a)



(b)

Figure 4.1: Analytical solutions of Kanatani algorithm applied to a moving planar surfaces

lar we perform the computation through the estimated essential parameter derived from Optical flow. The flow performed before does not give still a sufficient precision for evaluating multiple motion like complex rototranslation. We have seen in the analysis of the least mean squares that the aperture

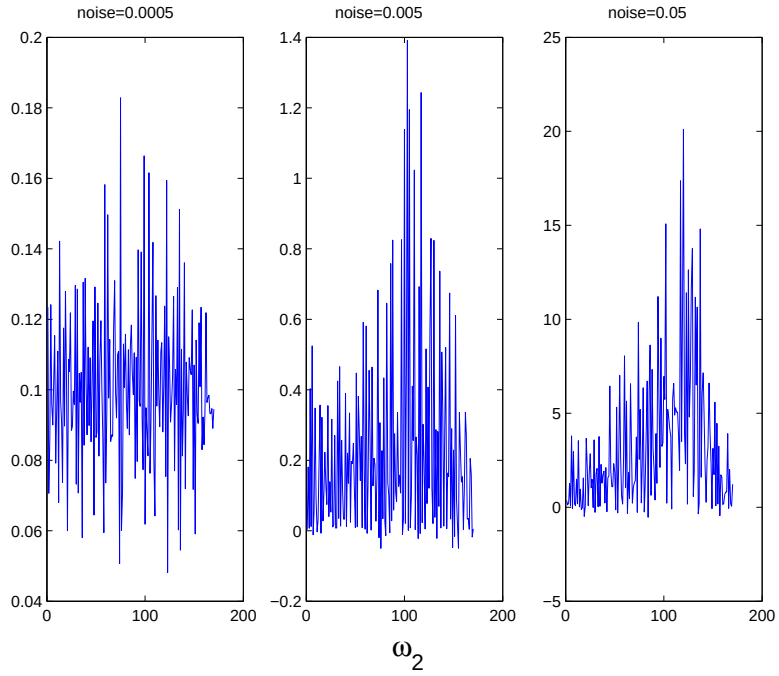


Figure 4.2: Impact of the noise level

problem and the field on the boundary of the objects introduce a significant error. In the Fig (4.5) we analysed a cube rotating in the Z axis and in the same time traslating in order to have the parameter c_3 different from zero. We have employed a chess pattern to avoid the aperture problem. In Fig (4.6) we have a particular region of the field.

Fig (4.7) shows the correct solution of the rotating plane. We choose to rotate it at three different velocities and we can see that effectivelly in the graph of ω_3 there are changes at 30 and 50 frames. Note that the relative velocities are different from zero as should be. Since we consider for the translation velocity bigger than the unity, this values can be filtered and we can assume to be null.

4.3.2 EKF

In this section we simulate the behavior of the extended Kalman filter. We start with a simple example in the 3D matlab simulator with a rototransla-

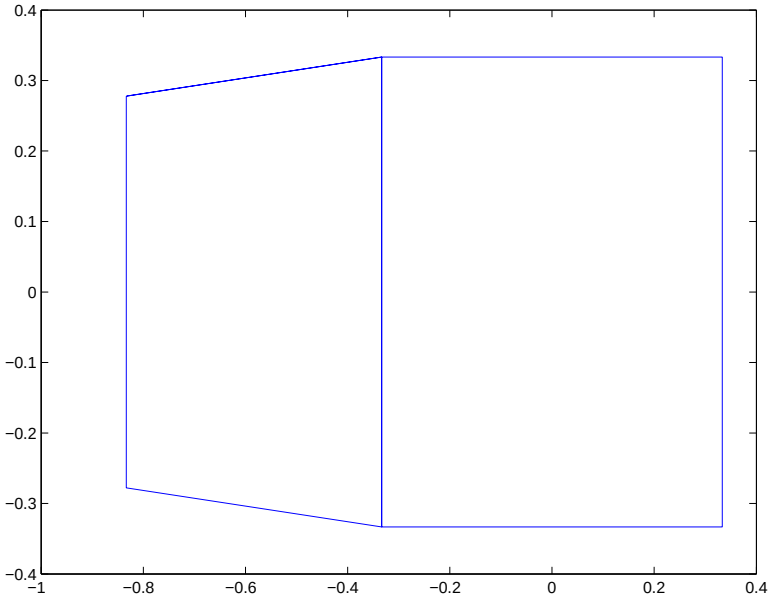


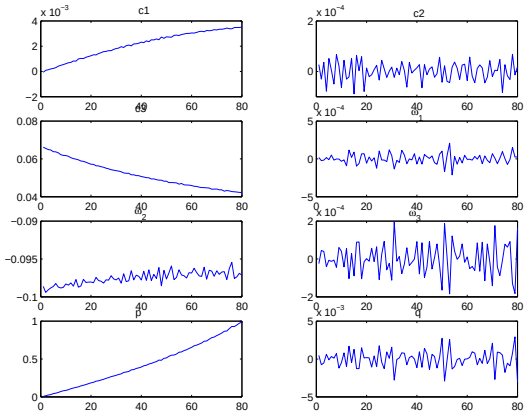
Figure 4.3: View of two moving planes connected

tion, following the protocols described in [8] and used in the previous analytical method. We would like to test two details: we expect that the algorithm tracks well the created synthetic moving planar surface in the camera plan and we want that in the same time it obtains the correct values of the shape parameters. In the simulated world, we deal with the possibility to introduce a controlled noise, as we have done previously. With different noise conditions, we adjust the parameters inside of the filter in order to achieve better performance. For this reason the considered estimation method results in higher ductility when comparing to the analytical method.

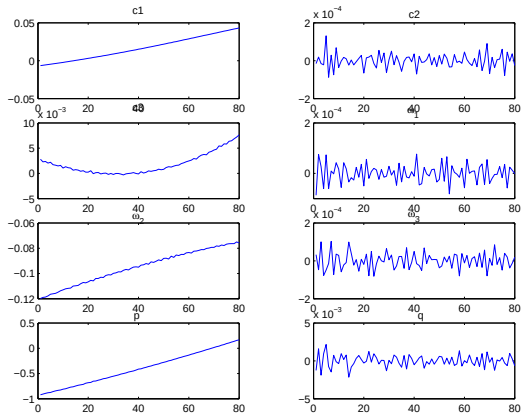
As in the previous example, we introduced the same rototranslation that derives from the parameters $(\omega_1, \omega_2, \omega_3) = (-0.2, 0.1, -0.1)$ and $(b_1, b_2, b_3) = (0.1, 0.1, -0.1)$. The rotation was applied around the same reference point used before $P_0 = (0, 0, 3)$. We inserted the Gaussian noise that we used in the analytical method, in order to compare the result.

In Fig (4.8), we show the result due to the tracking on the image plan from a corner of the surface. The different noise conditions have been studied.

In Fig (4.9) for two cases of noise we determinate the correct velocity value of ω_1.



(a)



(b)

Figure 4.4: Analytical solutions of Kanatani algorithm in the case of more planar surfaces

Finally, in Fig (4.10) we consider the planar shapes pq parameters compared to the real values.

The last part serves to apply the EKF into the virtual environment. It has been used the pattern describes in the chapeter 2 in Fig (3.2) in order to

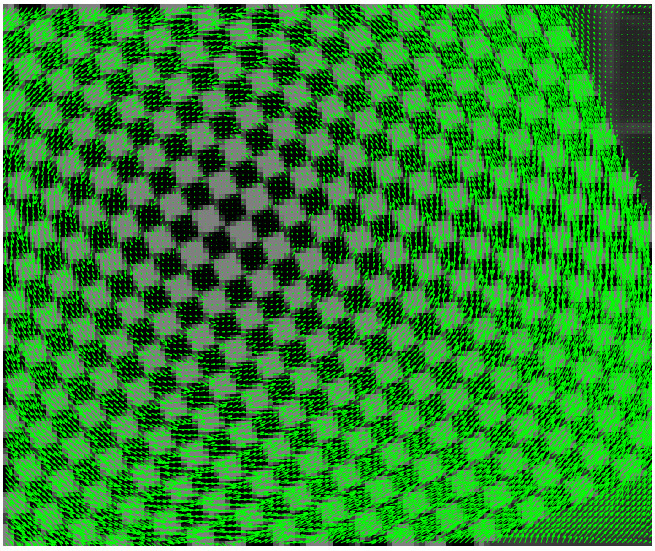


Figure 4.5: Rototraslation planar surface

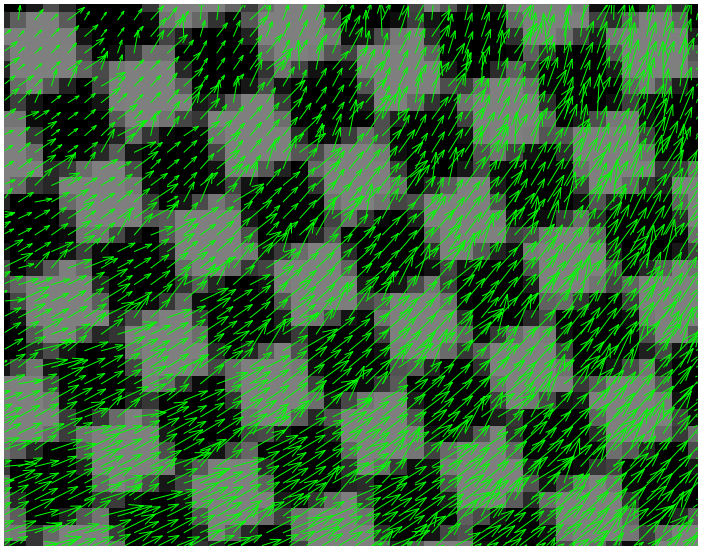


Figure 4.6: Region of a planar rotating surfaces

track the five points drawn on it. The motion that we capture, it is composed

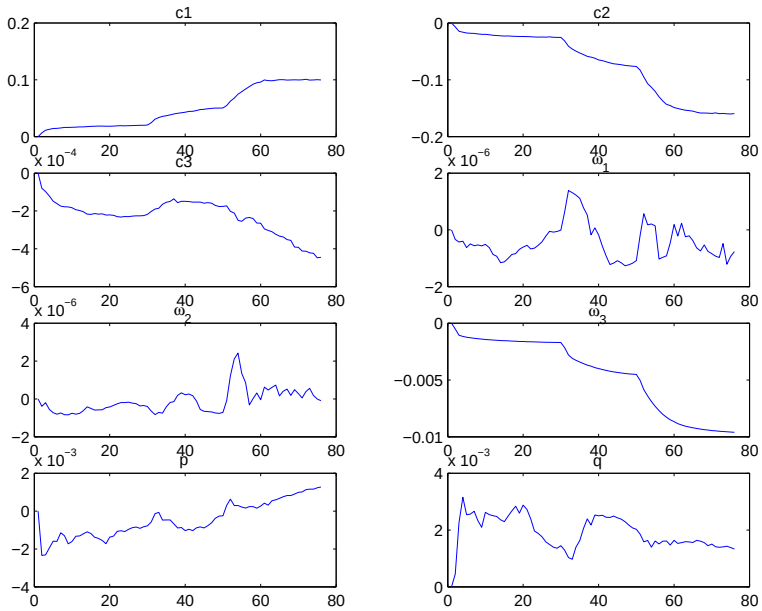


Figure 4.7: Solutions for RotoTraslating plane

by a rotation on the X and Y axes togheter with a translation in the Z axis.

In Fig (4.11) the tracking has been shown the motion of two features of the plane togheter with the tracking procedure. The tracking works fine.

Finally it has been determinated the evolution of the shape parameters over the time. In Fig (4.12) we show that the parameter pq are increasing.

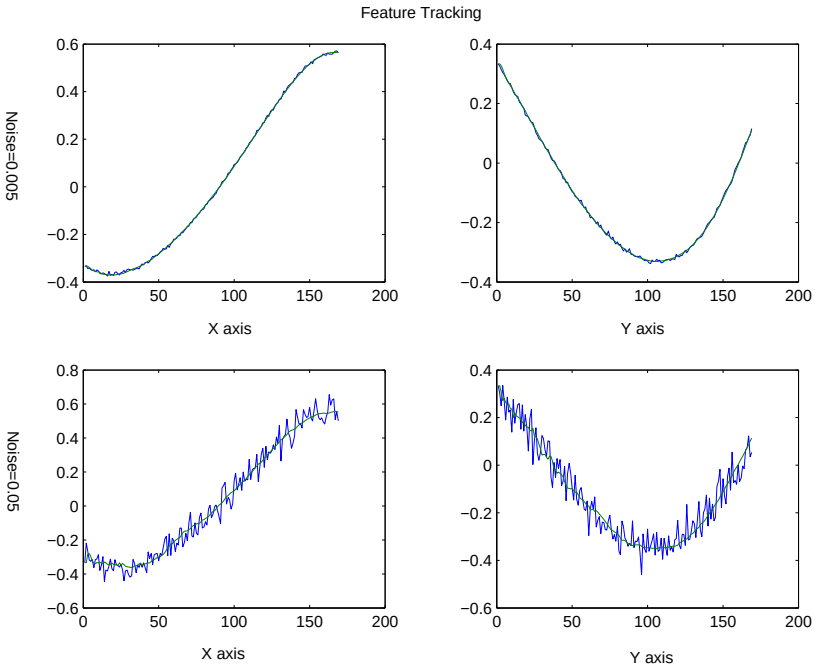


Figure 4.8: Tracking Features with EKF

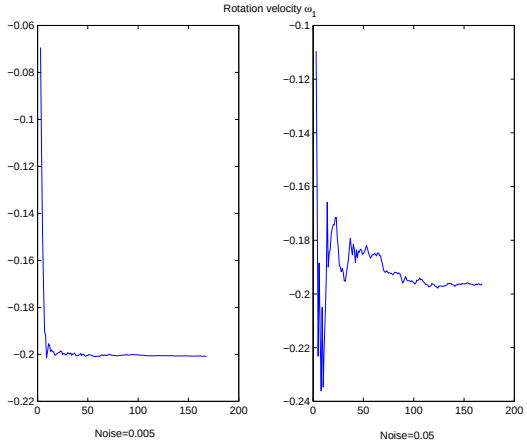


Figure 4.9: Velocity rotation

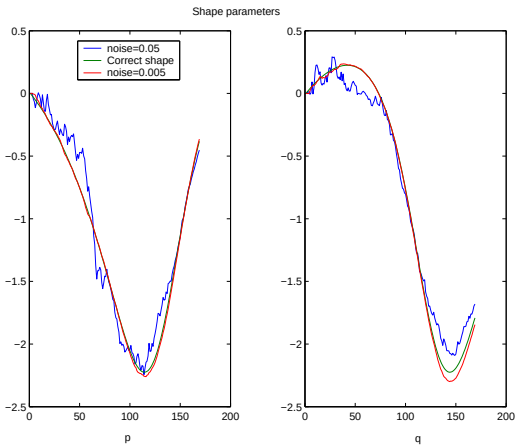


Figure 4.10: Estimation of the shape parameters

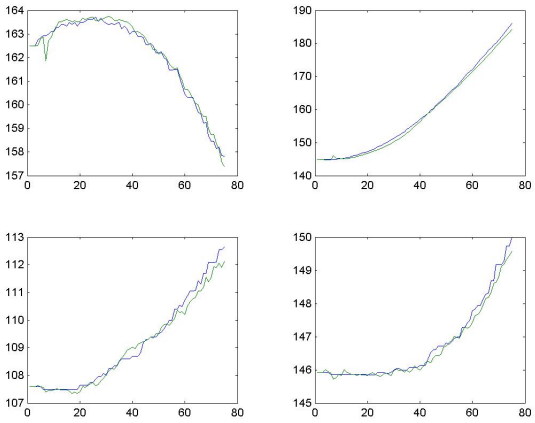


Figure 4.11: Tracking of features on the planar surfaces

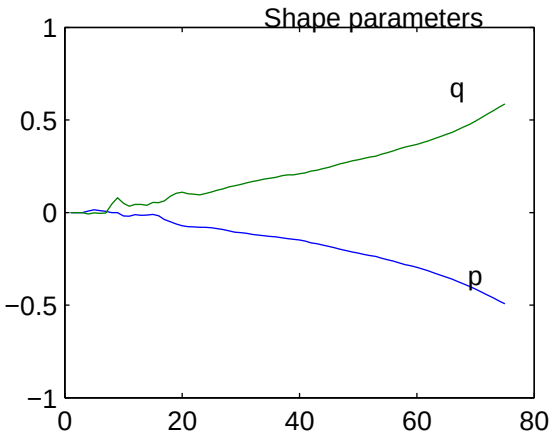


Figure 4.12: Tracking of shape parameters on the planar surface

Chapter 5

Conclusions

The subject of this thesis comprises the study of a framework for Dynamic Vision and in particular for the shape for motion case. We divided the estimation into two parts: image characteristic and objects characteristic estimation in order to perform the shape recovery. The case of moving planar surfaces has been studied. The reason of the planar choice derives from the fact that it is easier to determine its parameters. The model, as we have seen, is described by simple second order polinomia through eight essential parameters. Besides, the planar surface is the simplest 3D object that can be found in many human-made things.

In the first stage we inspected the problem to achieve information from the camera's image plan. In particular, we consider the general issue of dense computatation through an algorithm that has the duty to perform the optical flow estimation. We decided to choose the least mean squares algorithm proposed by [3] for some interesting characteristics suitable for a dynamic vision scheme. In fact, this adaptive recursive approach results in a fast and easy computation. Moreover, the estimation process refines the flow over the time guarantying good results.

We applied the algorithm into a simulated world and into the Virtual Robot Environment with different cases of pattern for investigated objects. For the case of chess board pattern, the estimation is successful, also for the reason that the aperture problem is limited. Unfortunately, in case of general patterns, this problem becomes stronger and the algorithms fail to recover the correct field. We also checked the method with a real movie. We took the famous Rubik cube sequence where the cube is fixed on a rotating platform. When the velocity field is stationary over the time and the pattern is not

colored the algorithm deals well with the task. However for the cube, where there are long black lines and big white areas, the least mean squares has serious estimation problem.

The tests that we carried out with the least mean squares algorithm include a tuning stage, as well. The behaviour of the parameters μ, β and the number of iterations were evaluated through a simple example. The step size, in particular, includes the propriety of stability and velocity of estimation and is the most important factor of the algorithm. The values of tuning depends on the different kind of considered scene. For this reason it is better to start the computation with an auto calibration of the parameters. The suggestion of using the bound that comes from the trace of the correlation matrix is not precise enough. Instead, for the step size μ, the NSD searching method gives a good initial conditions but since it requires the propagation of the covariance and correlation matrix, it is useful just for the first computation.

The fact that the estimation succeeds in case of stationary flow field and for irregular patterns is in part due to the reason that we propagate the state in time with the simple initial conditions. In [3] the authors used a second order AR model. Unfortunately we have not tested the possibility to use other models.

The second part that we have studied includes the shape from motion problem. In the thesis we consider a new approach that has been formulated in several papers due to Ghosh. The perspective system theory can be an interesting starting point for study in details a control system, inclusive a camera. The shape from motion is one of those exiting subjects. In this thesis we have studied the perspective system in order to understand the parameters, which are possible to be recovered. The recovery part has been analysed through two algorithms that have been cited by Ghosh.

The algorithm of Kanatani belongs to an algebraic approach and presents some drawbacks in case of noisy data, which is typical for this kind of methods. We tried to perform with this algorithm the shape parameters estimation through the optical flow calculated at the earlier stage of our investigation. After having obtained the essential parameters values we use the theorem due to Kanatani. Unfortunately, if the flow stage results are not precise, the algorithm does not perform well. A complex rototranslation with the precision of the algorithm cannot be recovered, in contrary to simple pattern and simple rototranslation. Thus we detected that the algorithm seems to be very sensible to the correctness of the data and a minimal distortion brings the framework to a failed estimation.

We confirmed that the extended Kalman filter, instead, does not have particular problem for noisy data and is more suitable for real application. Moreover, it has the interesting propriety to include the tracking of features on the image plane and thus performs image and objects characteristic estimation in the same time. Also when using the virtual environment the EKF works rather properly. The main problem for this kind of algorithm is that there is not a clear law for the convergence.

In general we have seen that the EKF algorithm performs better than the algebraic method and for real application is definitely the best choice.

5.1 Future work

Dynamic vision is a challenging area that has been studied in the last ten years. It is a young field where it is possible to develop interesting applications in control and robotics. In this thesis a framework to achieve results in this area has been originated and pointed out. However there are some aspects that we would like to leave for further studies.

- Some of the algorithm inside of this thesis has been tested through simulations and virtual environments. The natural continuation is to perform the presented methods inside of a real system. As an initial approach, we tried the case of the Rubik cube.

- The segmentation part, which has a task to fit parametric models inside of the estimated flow is not complete yet. We detected a simple moving plane in order to achieve the eight parameters. Extending the analysis for multiple motion and using a more complex segmentation algorithm is necessary to obtain good performance in order to use the framework in real conditions.

- The thesis presents the framework divided into different stages as for instance optical flow, segmentation and recovery. An interesting continuation for the future work is to consider an estimation scheme that connects the metioned fields in one single block of useful approach, what could result in development of many contemporary subjects.

  The optical flow part performs better if we know the extension of the objects and their dynamics. This means that the information of the segmentation and the recovery parts, that are successive stages of optical

flow estimation, can be used for improving the correct flow detection. Moreover, all these stages need a better flow detection.

- We have studied the dense computation scheme for recovering optical flow for the reason that it includes a more general information about the motion of the 3D world with respect to the sparse computation. The purpose is to detect the structure and the dynamics of the scene through general objects, thus extending the simple case of moving planar surfaces.

- We would like to study in details the perspective system theory with extension to the linear system case. It can be interesting in future to consider for instance controllability conditions in order to be able to use such application for general visual servoing and robotics

In conclusion basing on well-established data we tried to improve the problems of the Dynamic Vision field of study and thus to contribute to development of this subject. Moreover I believe that my work and few originated ideas will be continued and my interesting results will be successfully extended for more complicated cases. It is important to analyse carefully different aspects of Dynamic Vision in order to deal with its exciting contemporary problems.

Bibliography

[1] B.K.P.Horn. *Robot Vision.* The MIT press, Cambridge Massachusetts 1986

[2] K.I. Kanatani *Group-Teoretical Methods in Image Understanding* Springer Verlag 1990

[3] M. Elad & A. Feuler *Recursive optical flow estimation-Adaptive Filtering approach* Vol. 9,No 2, June, pp. 119-138, 1998

[4] S.Haykin *Adaptive Filter Theory* Prentice-Hall New York 1986

[5] B.K. Ghosh & E.P.Loucks *A Perspective Theory for Motion and Shape Estimation in Machine Vision*, SIAM Journal on control and optimization, vol 35, 1995

[6] B.K. Ghosh & E.P.Louck *A realization theory for perspective system theory and its application to machine vision*, IEEE Transaction on Automatic Control, vol 41, 1996

[7] B.K. Ghosh & J.Rosenthal, *A generelized Popov-Belevithc-Hataus test of observability*, IEEE Transaction on Automatic Control, vol 40, 1995

[8] H.Kano & B.k.Ghosh & H.Kanai *Single camera based motion and shape estimation using extended Kalman filtering*, Mathematical and Computer Modelling, vol 34, 2001

[9] R.M. Murray & Z.Li&S.S.Sastry *Robotic Manipulation* CRC Press 1994

[10] E.Trucco & A.Verri *Introductory techniques for 3-D computer vision* Prentice Hall 1996

[11] S.S. Beauchemin & J.L. Barron *The computation of Optical Flow* 1995

[12] M.J. Black *Robust Incremental Optical Flow* PhD thesis, Yale University, New Haven 1992

[13] A. Blake & A. Zizzerman *Visual reconstruction* The Mit Press, Cambridge, Massachusetts, 1987