A Geometric Framework for Dynamic Vision

Thesis by

Stefano Soatto

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1996

(Defended May 17, 1996)

to Anna and Arturo

Acknowledgements

My thanks go first to my supervisor, Pietro Perona, for his open-minded and creative interaction during the past four years at Caltech. Members of my (various) committees provided me with helpful advice on the material of the thesis: Joel Burdick, John Doyle, Christof Koch, Jerrold Marsden, Manfred Morari, Richard Murray and Jean Ponce. Ruggero Frezza and Giorgio Picci supported with enthusiasm my early efforts in this area at the University of Padova. During my visits to various laboratories around the country, I had the pleasure to interact with researchers of broad and far-reaching visions such as Ruzena Bajcsy, Roger Brockett, Ernst Dickmanns, S. Y. Kung, Sanjoy Mitter and Shankar Sastry. I also enjoyed inspiring conversations with other researchers that are active in the field of motion vision, such as Ali Azarbayejani, Bijoy Ghosh, John Oliensis, Harpreet Sawhney and Eero Simoncelli. Caltech is an extremely fertile environment thanks to the friendly atmosphere that allows a high level of interaction among faculties and students. I had the luck to share my research puzzles with fellows such as Andrea Mennucci, Willem Sluis, Francesco Bullo and Michiel von Nieuwstadt. Within Caltech, the Vision group is a keen example of such peculiar features; thanks to Jean-Yves Bouguet, Jochen Braun, Michael Burl, Enrico Di Bernardo, Luis Goncalves, Christopher Kolb, Mario Munich, Jennifer Sun, Marco Tartagni, Joe Weber and Markus Weber. Finally, I wish to thank the faculty of the department of Mathematics and Information Sciences at the University of Udine, for their support during my leave of absence: Adriano Pascoletti, Vito Roberto, Goffredo Pieroni and Furio Honsell.

The research in this thesis has been supported by the NSF Center for Neuromorphic System Engineering; grants from ONR, NSF and scholarships from the University of Padova and the A. Gini foundation.

Abstract

This thesis explores the problem of inferring information about the three-dimensional world from its projections onto a camera (images). Among all visual cues, we do not address "pictorial" ones, such as texture or shading. Instead, we concentrate on "dynamic" cues, which are associated with variations of the image over time.

In order to eliminate pictorial cues, one may represent the world as a collection of geometric primitives, such as points, curves or surfaces in three-dimensional space. Then, from the two-dimensional motion of the projection of such primitives onto the image, one can infer the three-dimensional structure of the world and its motion relative to the viewer.

"Three-dimensional structure from two-dimensional images" has now been a central theme in Computer Vision for over two decades, and tools from Linear Algebra and Projective Geometry have been widely employed to attack the problem as a "static" task. It is only in recent years that the role of *time* has started to be recognized, after the influential work of Dickmanns and his coworkers on vehicle guidance on freeways.

We do not impose restrictions on the structure of the environment, and we cast the problem of *general* three-dimensional structure and motion estimation within the framework of Dynamical Systems. We show how different algebraic constraints on the image projections can be interpreted as nonlinear and implicit dynamical models whose (unknown) parameters live in peculiar differentiable manifolds that encode three-dimensional information. Recovering such three-dimensional information then amounts to identifying dynamical models while taking into account the geometry of the parameter manifolds.

Contents

List of Figures

Chapter 1 Introduction

Vision is a remarkably powerful sense that animals exploit to interact with their environment. A single still image of a scene is already a rich source of information on three-dimensional structure, combining different cues such as texture, shading, contours, cast shadows etc. (see figure 1.1). Such cues, however, are

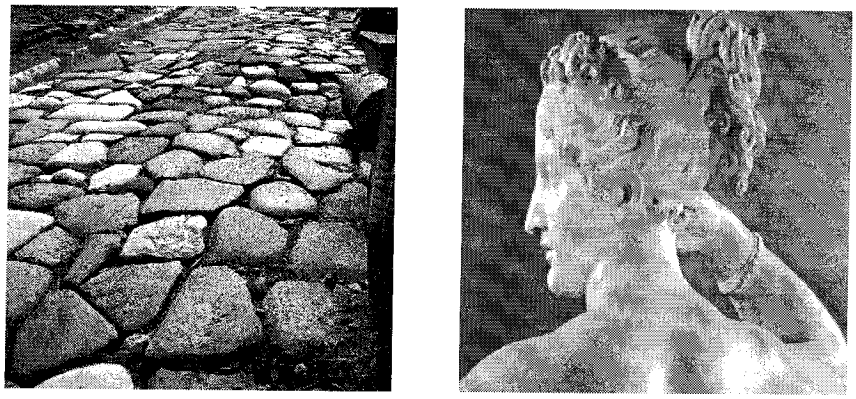

Figure 1.1: *Some "pictorial" cues for three-dimensional structure: texture (left), shading (right).*

intrinsically ambiguous since an image is a projection of the three-dimensional world onto a two-dimensional surface, and therefore a whole dimension is lost. Indeed such ambiguities are systematically exploited, for instance, in the entertainment industry to produce visual illusions.

Two images of the same scene taken from different viewpoints can be put in correspondence and provide an additional cue, called "stereo" (figure 1.2). Stereo is exploited by the human visual system to infer the "depth" structure of the scene in the close-range. More generally, if we consider a stream of images taken from a moving viewpoint, the two-dimensional image-motion[1] can be exploited to infer information about the three-dimensional structure of the scene as well as its motion relative to the viewer.

That image-motion is a strong cue is easily seen by eliminating all pictorial cues until the scene reduces to a cloud of points. A still image looks like a random collection of dots but, as soon as it starts moving, we are able to perceive the three-dimensional shape and motion of the scene (see figure 1.3).

[1]We use the improper diction "image-motion" or "moving image" to describe the time-change of the image due to a relative motion between the scene and the viewer.

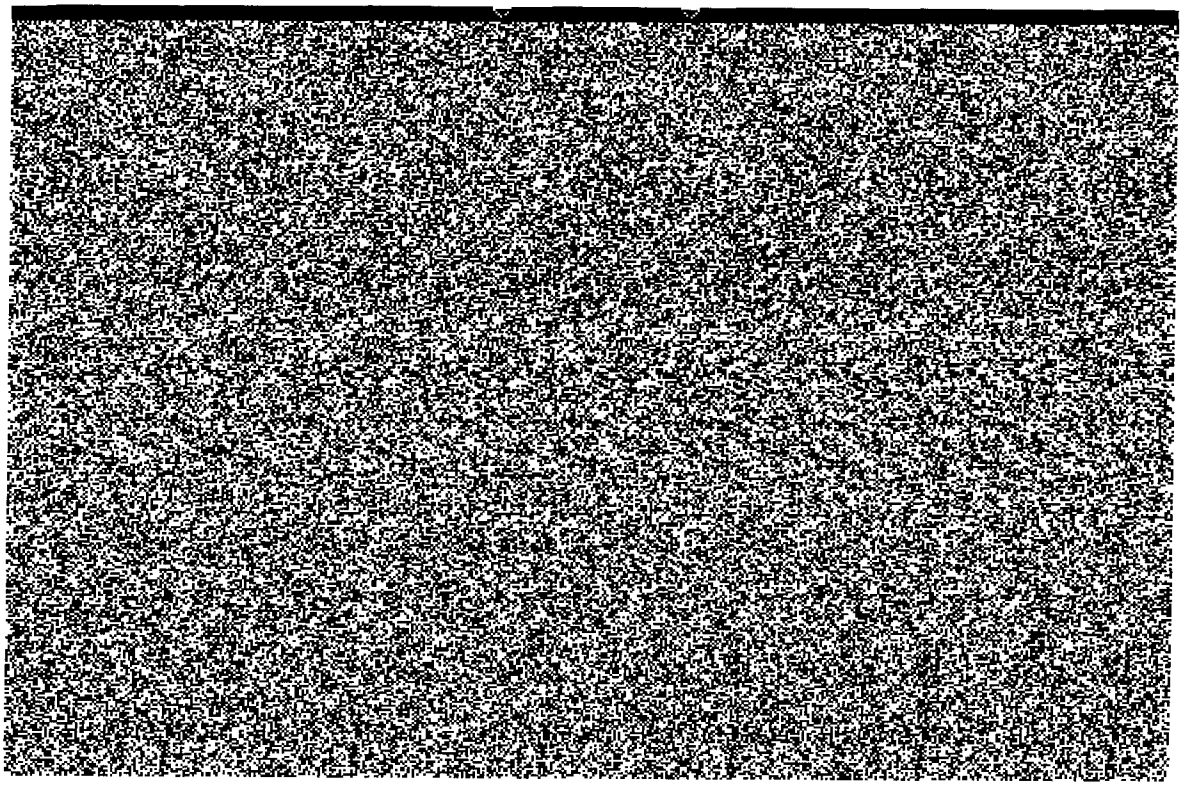

Figure 1.2: *Stereo as a cue in "random dot stereogram". When the image is fused binocularly, it reveals a "depth structure". Note that stereo is the only cue, as all pictorial aspects are absent in the random dot stereograms.*

As the words suggest, "dynamic vision" deals with images that change over time, for instance under the action of the viewer's motion or due to changes in the shape of the environment. The term "motion vision" is also used in contrast to "pictorial vision" which deals with the analysis of static images. In this thesis we will concentrate on motion vision, and study how image-motion can be exploited as a cue to infer the three-dimensional structure and/or motion of the scene.

Our prototypical problem is then

> *from a sequence of moving images*[1]*, estimate the three-dimensional structure and/or the relative motion between the scene and the camera.*

There are two concepts that are central to our approach to motion vision: *dynamical systems* and *geometric invariance*. Under some assumptions about the structure and motion of the scene, it is fairly natural to cast motion vision within the framework of *dynamical systems*. In fact, the scene can be described as a point on

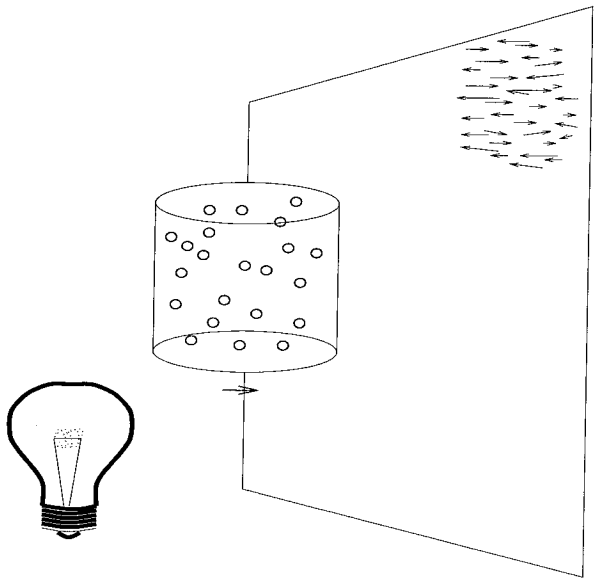

Figure 1.3: *2-D image-motion is a cue to 3-D scene structure: a number of dots are painted on the surface of a transparent cylinder. An image of the cylinder, which is generated by projecting it onto a wall, looks like a random collection of points. If we start rotating the cylinder, just by looking at its projection we can clearly perceive the existence of a three-dimensional structure underlying the two-dimensional motion of the projection.*

some geometric space, which moves under the action of motion, therefore describing the state of a dynamical model. The output of such a model is the *image* of the scene, which depends upon its structure and motion, but also upon other parameters such as brightness, color, reflectance or illumination of the scene.

Depending upon the particular application, one may be more interested in some parameters or in others. For instance, if we want to build a CAD model of some object, we do not care about its particular pose and motion when we first look at it. Therefore, in such a case we are interested in shape, but not in pose or motion. If we want to be able to render the object for visualization, we may also need to reconstruct the reflectance properties of its surface. On the other hand, when we drive and want to follow a prescribed path, we do not want our control to depend upon the particular landscape we happen to be driving through.

Here we face a dilemma: we are interested in the state of a dynamical model whose measurements happen to depend upon other unknown parameters, which may not be interesting per se. What strategy should we pursue? Should we try to estimate all unknowns, and then just retain whatever we are interested in, or should we rather try to devise models for the interesting parameters, that are "invariant"[2] with respect to

[2]Such an invariance may be achieved, as we will see, either by a pre-processing of the data (such as feature-tracking to get rid of all illumination-related parameters) or by appropriate representation of the space of unknown parameters.

uninteresting ones?

We will not try to give a general answer. Rather, we will discuss both alternatives in relation to specific domains of application of dynamic vision. We will first study the simultaneous estimation of three-dimensional structure and motion, and then explore various strategies to design models that are invariant to some of the unknown parameters.

In the same fashion as vision is a crucial sense for animals to successfully interact with each other and with their environment, we feel dynamic vision will play a crucial role as a sensor in control systems with autonomous capability. For this to happen, estimates of three-dimensional structure and/or motion need to be performed in real-time in a *recursive* and *causal* fashion. While the early involvements of vision as a sensor were mostly at low-level in restricted environments, more complex applications require multi-level control structures that exploit visual information in a hierarchical way in order to perform low-level control task (such as keeping the center of the lane while driving on the freeway) as well as higher-level decisions (such as changing lanes, passing slower cars, detecting unsafe behaviors of neighboring drivers). There are already in the literature examples of successful applications of vision in the loop of control systems at different hierarchical levels. In particular, Dickmanns and his group [27] have equipped a fleet of full-size vehicles capable to drive on public freeways with normal traffic at speeds up to 150 Km/h while reading speed signs, switching lanes, passing slower cars etc.. As the hardware improves, vision also starts being acknowledged as a sensor for *control systems* involved, for instance, in robotic manipulation, tracking, docking, surveillance etc. [10, 42, 43, 49].

1.1 Relation to previous work

Humans have always been extremely effective at estimating the shape of the environment and its relative motion, and they use such a proficiency for facing everyday tasks such as walking, avoiding obstacles, grasping objects. The mathematical tools for formalizing the problem, namely the geometry of the Euclidean group and perspective projection, have been available for a long time. It may therefore come as a surprise that the problem of estimating shape and motion from images has been formalized only in recent years [103], despite the fact that one could find the problem "in nuce" in the early work of von Helmholtz [105] and that of Gibson et al. [38]. Although the problem of estimating three-dimensional structure and motion has

now been a major theme in Computer Vision for over a decade, it is indeed far from being solved, and the literature still appears as a collection of apparently unrelated methods and schemes.

In 1981 Longuet-Higgins introduced a bilinear constraint on the image coordinates of an object onto two different views, involving the motion parameters in a non-linear fashion [73]. His work has triggered a stream of research known under the name of *epipolar geometry* which has been extended and made popular, among others, by Faugeras (see [30] for an overview). The literature on 3-D shape and motion estimation from two or more views has grown considerably since then, and we do not attempt to give an exhaustive coverage here. We only report a recent result of Faugeras, who has characterized all bi-linear, tri-linear and quadri-linear independent constraints involving image projections, motion and calibration parameters [31]. Such constraints are interpreted as polynomial equations whose coefficients are (nonlinear) functions of the motion parameters. The process of finding roots to such algebraic equations (usually in the complex field), and that of extracting motion parameters from the coefficients, can hide the geometric structure of the parameters to be estimated, and makes it difficult to address the issue of how to treat the inevitable measurement errors (see [108]).

In this thesis we take a different approach: rather than seeking algebraic constraints on shape and motion parameters, we cast the problem of structure and motion estimation within the framework of dynamical system state estimation. The feasibility of the problem of structure and motion estimation can then be studied as the observability of the corresponding dynamical models. The observability space plays the role of a *constraint generator*, and observers are used in order to *"solve"* such constraints in an incremental (recursive) and causal fashion.

It was not until a decade ago that the role of *time* started to be recognized in the Computer Vision community. Gennery [36] and Dickmanns [28] were the first to use Extended Kalman Filters in order to recover shape and/or motion parameters from streams of images, although in more restricted situations (known 3-D shape in the case of Gennery, partially structured environment in the case of Dickmanns). Recently, quite a few schemes that use local observers for estimating 3-D shape and/or motion have appeared in the literature (see for instance [5, 15, 46, 78, 84]). Dayawansa et al. [24] and Ghosh et al. [37] addressed the issue of *observability* and derived a test for perspective systems (linear systems with measurements in projective spaces) which resembles the familiar Popov-Belevitch-Hautus rank test for linear systems. Schemes

arising from the framework of perspective systems, however, have proven effective so far mainly on *planar* curves or surfaces [37], when the transformation induced on the image plane is an homography and, therefore, it can be represented as a linear transformation of the homogeneous coordinates.

Contribution of the thesis

The first observation is that the (well-known) setup of epipolar geometry can be translated into a dynamical context by *viewing the essential constraint as a dynamical model.* Such a dynamical model has a peculiar form that turns out to be somewhat general, as many other algebraic constraints involving images and 3-D motion parameters can be interpreted as dynamical systems of the same form (for instance the "Subspace constraint" or the "Plane-plus-parallax" setup). The thesis proposes therefore a *unifying framework* that allows deriving different constraints by following simple principles from the theory of dynamical systems (for instance that of "observer reduction"). Moreover this framework allows one to generate *novel* constraints simply by choosing a fixation function. Different models are characterized by different geometry of the space of unknown parameters.

Once we have cast the problem within the framework of dynamical systems, the problem of inferring three-dimensional information resorts to identifying the parameters of (nonlinear and implicit) dynamical models. In order to perform the estimation properly one must take into account the geometry of the parameter manifolds. We have observed that the so-called "Essential space" – which has been previously characterized in the complex field as the set of zeros of certain algebraic equations and was proven to contain singular points – is indeed a *differentiable manifold* of dimension six, which coincides with the tangent bundle of the rotation group. We have proposed two methods to design observers on such a peculiar manifold. Other schemes can be obtained by changing the parameter space onto different sections (slices) of the essential manifold, thus generating a *geometric stratification.*

It is possible, within this framework, to talk about issues such as *outlier rejection, segmentation, self-calibration, motion control* in a fairly natural manner. We can also perform a thorough *experimental comparison*, since the schemes differ only by the ir local coordinate representation, while the estimation setup remains unchanged.

1.2 Reading the thesis

This thesis is divided into three parts. Part III contains background material and can be used as a reference. In particular, Appendices A and B describe the process of tracking point-features and calibrating a camera, which are necessary steps towards three-dimensional shape and motion estimation. It may be skipped by assuming that we are given a method that solves the so-called "correspondence problem". The most critical readers will argue that the correspondence problem is indeed very difficult to solve satisfactorily. We agree, and indeed this observation is one of the central motivations for chapters 3 to 7, where we present models that integrate visual information over time even in the presence of "miopic" (local) feature tracking.

Part I contains the core material of the thesis. We start from the basic constraints that define the problem of "Structure from Motion" and see how they can be employed to simultaneously estimate three-dimensional structure *and* motion (chapter 2). The feasibility of such a problem is the subject of chapter 4. We then explore alternative strategies to render the basic models invariant with respect to some of the unknown parameters. In chapter 3 we exploit the framework of "epipolar geometry" in a dynamical context to decouple structure from motion. In chapter 5 we further decouple the direction of translation from the rotational velocity. Chapter 6 shows how it is possible to decouple states that are affected by the so-called "bas-relief ambiguity".

The methods presented in chapters 3, 5 and 6 rely upon being able to explicitly isolate the parameters to be eliminated from the observability space. While this is not always possible, one can do so "implicitly" by enforcing "fixation constraints" (chapter 7).

Chapter 8 deals with the problem of segmentation, and shows how it is possible to detect outlier measurements coming from the feature tracking, thus rendering the schemes proposed robust to the inevitable tracking errors. Chapter 9 shows how it is possible, in principle, to estimate calibration parameters along with three-dimensional motion. Chapter 10 shows a simple application of structure-independent motion estimation to motion control.

Each chapter contains an experimental section to highlight the main features of the schemes proposed.

Part II consists of a series of simulation experiments that compare the performance of all schemes presented in this thesis. In order to render this part self-contained, the main points of Part I are summarized in the introductory section of chapter 11.

Finally, in chapter 12 we discuss some directions of future development.

Part I

Visual Motion and Structure Estimation

Chapter 2 Modeling Structure From Motion

In this chapter we formalize the problem of "Structure from Motion", which consists of estimating the three-dimensional structure *and* motion of a scene from the two-dimensional image-motion. We concentrate on portions of the scene which *move rigidly* relative to a viewer who measures the *perspective projection* of the scene onto the retina (or image-plane). We will see that the images of a moving scene depend not only upon its shape, but also upon its pose and motion, which are unknown. We derive dynamical models whose state includes all unknown parameters, that could therefore be estimated by means of state observation.

Background and notation for the chapter

We represent the scene as a collection of point-features in three-dimensional Euclidean space, and we assume that we can measure their perspective projection onto an image-plane over time (the so-called "correspondence problem"). A method for solving the correspondence problem is outlined in appendix A. We also assume that we know the internal geometry of the imaging device (calibration, see appendix B).

A careful reader will argue that the methods available for performing feature-tracking are intrinsically local in nature. Therefore, although it is easy to solve the correspondence for small displacements and short sequences of images (fewer than 10 frames in most practical situations), it is virtually impossible for local methods to maintain track of features over extended periods of time. This is indeed one of the central motivations of the thesis: the brightness-constancy constraint is intrinsically local, and therefore whatever algorithm uses it ought to be able to cope with its limitations. We will present methods that allow us integrating visual information over long periods of time, even when each single feature has a very short life-span. In the limit we can assume that each feature survives only between pairs of frames (the so-called "optical flow") as an approximation of the projection of the velocity of points in three-dimensions.

This is the reason why we find that the performance of standard feature-tracking or optical-flow algorithms, as they have been available for more than a decade, is good enough for our purposes.

We will use extensively the properties of rigid motions [81]. Note that it makes no difference whether it is the viewer or the scene moving. All that matters is the relative motion between the two. We will use the exponential both as a local-coordinate representation of a rotation matrix R through a rotation vector Ω, and to compute the integral of a (piecewise) constant rotational velocity ω. The most frequently used symbols in this chapter are

- $\mathbf{X} \in \mathbb{R}^3$: three-dimensional Euclidean coordinates of a point
- $\mathbf{x} \in \mathbb{R}\mathrm{P}^2$ or $\mathbb{R}^2$: perspective projection of the point $\mathbf{X}$. It can be expressed in homogeneous coordinates (three scaled numbers) or plane coordinates (two numbers), depending upon the context
- $n \in \mathcal{N}(0, \Sigma_n)$ zero-mean Gaussian measurement noise. When the measurements are encoded in homogeneous coordinates it is intended that the noise only affects two independent directions (not the scale component).
- $g \in SE(3)$ a rigid motion, composed of:
- $T \in \mathbb{R}^3$, a translation vector, and

- $R \in SO(3)$, a rotation matrix.
- $v\wedge \in se(3)$ is the instantaneous generalized velocity corresponding to $g \in SE(3)$. It is related to g via the exponential, and composed of:
- $V \in \mathbb{R}^3$, a translational velocity vector, and
- $\omega \in \mathbb{R}^3$, a rotational velocity vector.

Outline of the chapter

In section 2.1 we formalize the notion of "shape", which is then used in section 2.2 in order to define models for shape, pose and motion having a projection in the measurement model. We have implemented some of these models, and we discuss their qualitative properties in section 2.3. In sections 2.4 and 2.5 we motivate the use of reduced models where different states are decoupled in the dynamics of the observer. Such issues are then discussed in detail in the subsequent chapters 3-6.

2.1 Shape Spaces

Let $P^i \in E^3;\ i = 1 \dots N$ be points in three-dimensional Euclidean space, and let $\mathbf{X}^i \in \mathbb{R}^3;\ i = 1 \dots N$ be their coordinates relative to some (orthonormal) inertial reference frame. We call $\bar{\mathbf{X}}$ the matrix that collects the coordinates of each point

$$\bar{\mathbf{X}} = \left[\begin{array}{ccc} \mathbf{X}^1 & \dots & \mathbf{X}^N \end{array} \right] \in \mathbb{R}^{3\times N} \tag{2.1}$$

which we will refer to as one *configuration* of the points P^i. Note that a change of the reference frame will alter the coordinates, although the underlying points remain the same. The concept of *shape* captures precisely

> *what is left of the configuration after the effects of translation, rotation and scaling have been factored out [60, 65].*

We describe the effects of translation, rotation and scaling as a *group action* on the set of configurations. To this end, let $g(t) \in SE(3)$ be a rigid motion, which acts on the P^i generating a trajectory on $E^3 \times \dots \times E^3$:

$$P^i(t) \doteq g(t)P^i \ \forall\ i = 1 \dots N;\ \forall\ t \tag{2.2}$$

or, in coordinates,

$$\bar{\mathbf{X}}(t) \doteq R(t)\bar{\mathbf{X}} + T(t)[1 \dots 1] \in \mathbb{R}^{3\times N};\ \forall\ t \tag{2.3}$$

where $R(t) \in SO(3)$ is an orthonormal rotation matrix which represents the orientation of the reference frame at time 0 relative to the one at time t, and $T(t) \in \mathbb{R}^3$ is a translation vector that represents the coordinates of the origin of the reference frame at the initial time relative to that at time t. Geometrically, the trajectories obtained under the action of rigid motion (translation and rotation) and scale can be described as a *fiber bundle* [40] (see figure 2.1). Each configuration represents an point on a *fiber* ϕ, which we can move along the fiber by rotation, translation and scaling:

$$\phi = \{\lambda(R\bar{\mathbf{X}} + T[1 \dots 1]) \mid \lambda \in \mathbb{R}, T \in \mathbb{R}^3, R \in SO(3)\}. \tag{2.4}$$

Each fiber is an equivalence class that encodes one single shape, which can therefore be represented a particular element of the class. To this end we may define a *projection* from the fibers onto the *base-space* of the bundle by "undoing" translation, rotation and scaling. We then define the "shape" of a configuration as the projection onto the base-space, which is therefore called the "Shape Space".

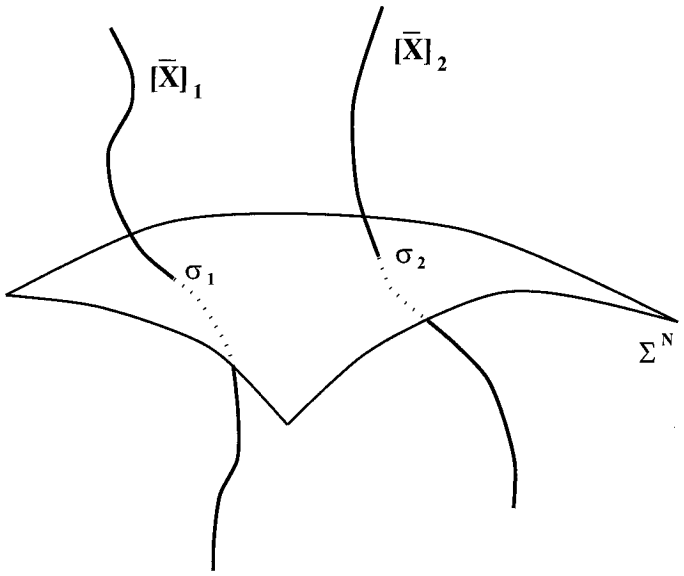

Figure 2.1: *The Shape Bundle: configurations are points on the fibers, which are projected onto the base-space to give a shape.*

Projection of the fibers onto the base-space and local coordinatization of Shape Spaces

Given a configuration, represented as a point on a fiber and encoded by a $3 \times N$ matrix, we wish to find the corresponding shape, which is the projection of the fiber onto the base-space. To do so, we need to rule out the effects of translation, scaling and rotation. This is done respectively by centering and normalizing the configuration, and then taking its modulo in SO(3).

Centering

Let $\bar{\mathbf{X}} \in \mathbb{R}^{3\times N}$ be a configuration matrix. Define the *centroid* of the configuration as:

$$\mathbf{X}^c \doteq \frac{\sum_{i=1}^{N} \mathbf{X}^i}{N} \tag{2.5}$$

and let $\mathbf{X}_c^i \doteq \mathbf{X}^i - \mathbf{X}^c$ be the centered points, which are the columns of the *centered configuration* matrix

$$\bar{\mathbf{X}}_c \in \mathbb{R}^{3\times N}. \tag{2.6}$$

Note that the elements of the centered configuration are not free, for the (right) null-space of the matrix $\bar{\mathbf{X}}_c$ must contain the vector $\mathbf{1} \doteq [1 \dots 1]^T$ of the appropriate dimensions:

$$\bar{\mathbf{X}}_c \mathbf{1} = 0. \tag{2.7}$$

In order to remember that the elements of $\bar{\mathbf{X}}_c$ must satisfy the above constraint, we will write

$$\bar{\mathbf{X}}_c \in < \mathbf{1} >^{\perp}. \tag{2.8}$$

Centering the configuration around the centroid has the effect of eliminating translation. In fact, all configurations that are translated with respect to a given one have the same centered-configuration matrix.

Column defection

In centering the configuration, we have introduced one constraint on the elements of the matrix $\bar{\mathbf{X}}$, namely the fact that its right null-space must contain the vector $\mathbf{1}$. Instead of carrying along a $3 \times N$ matrix with three constraints on its elements, it is more convenient to reduce it to a matrix with $3(N-1)$ *free* elements. In order to do so, we want to construct a map

$$\bar{\mathbf{X}} \in < \mathbf{1} >^{\perp} \subset \mathbb{R}^{3\times N} \longrightarrow \bar{\mathbf{X}} \in \mathbb{R}^{3\times(N-1)}.$$

Geometrically, we want to rotate (from the right) the configuration until the vector $\mathbf{1}$ coincides with its last column. In doing so we do not want to alter the structure of the configuration; therefore, we seek for an orthonormal $N \times N$ matrix $K \in O(N)$ such that

$$K \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = \frac{\mathbf{1}}{\sqrt{N}}$$

for, if $\bar{\mathbf{X}}_c$ is such that $\bar{\mathbf{X}}_c \mathbf{1} = 0$, then

$$\bar{\mathbf{X}}_c \mathbf{1} = \bar{\mathbf{X}}_c K \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \sqrt{N} = 0$$

which translates into

$$\mathbf{X}_{co}^{N} = 0 \tag{2.9}$$

where we define

$$\bar{\mathbf{X}}_{co} \doteq \bar{\mathbf{X}}_c K. \tag{2.10}$$

Once this is done, we can just delete the last column of $\bar{\mathbf{X}}_{co}$ and be left with a $3 \times (N-1)$ matrix of free elements

$$\bar{\mathbf{X}}_{co} \in \mathbb{R}^{3\times(N-1)}.$$

The task boils down to finding a matrix in $O(N)$ which has the last column equal to $\mathbf{1}/\sqrt{N}$. One instance of such a matrix can be found in [60]. Note that there are many of them and therefore the transformation of the centered configuration is *not canonical.* Once any such K is given, all the others that are of the form

$$K \begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix} \quad \forall\, k \in O(N-1). \tag{2.11}$$

Normalization

We now want to rule out the effect of scaling, by defining a *normalized-configuration.* This is simply done by projecting the configuration onto a sphere:

$$\bar{\mathbf{X}}_s \doteq \frac{\bar{\mathbf{X}}}{\|\bar{\mathbf{X}}\|} \in \mathbf{S}^{3N-1} \tag{2.12}$$

where the denominator is a suitably chosen norm. After centering and normalization, we are left with all points of the form

$$\bar{\mathbf{X}}_{cos} \in \{W \in \mathbb{R}^{3\times(N-1)} \mid \|W\| = 1\} \tag{2.13}$$

which belong to the set $\bar{\mathbf{X}}_{cos} \in \mathbf{S}^{3(N-1)-1}$.

Mod-out rotation

After scaling and translation, all possible (centered-scaled) configurations are rotated versions of the same "shape", which is represented as the equivalence class

$$[R\bar{\mathbf{X}}_{cos}]_{R\in SO(3)}.$$

Therefore, the *Shape Space* [1] is defined as

$$\Sigma^N \doteq \mathbf{S}^{3(N-1)-1}/SO(3). \tag{2.14}$$

In order to choose one element of the equivalence class one may perform the Singular Value Decomposition (SVD) of a scaled-centered configuration:

$$\bar{\mathbf{X}}_{cos} = RSW^T \tag{2.15}$$

where $R \in SO(3)$, $S = \text{diag}\{1, \sigma_2, \sigma_3\}$, and W has three orthonormal columns. The shape σ is then just the product $\sigma = SW^T$. The two scalars σ_2, σ_3, along with a local representation of the three orthonormal $N-1$-dimensional vectors $W_{.1}, W_{.2}, W_{.3}$ can be chosen as a local coordinatization of the Shape Space.

Remark 2.1.1 *The local coordinatization just described is based upon centering the reference frame in the centroid of the points, setting the unity to be the norm of the coordinates, and then rotate it so as to orient the first axis along the "largest" size of the cloud of points, then orienting the second along the largest size in the orthogonal complement.* If the cloud is isotropic, such re-orientation is ill-defined, *and so is the corresponding coordinatization of shape [60].*

Remark 2.1.2 *One major concern in using Shape Spaces in Structure from Motion is* occlusion. *In fact, suppose that we are given a configuration, and we want to compare its shape with the shape of the same configuration* where a point has been deleted. *The theory of Shape Spaces does not come at hand for this problem, since shapes with different numbers of points belong to different Shape Spaces altogether, and therefore they cannot be compared in a straightforward manner.*

In this chapter we will restrict ourselves to the case in which we have $N > 3$ *unoccluded* points and shapes have distinct singular values: $\sigma_1 = 1 > \sigma_2 > \sigma_3 > 0$. We will see in later chapters (3 and 5) alternative formulations that allow dealing with occlusions in a more principled way.

[1] The term "Shape Space" is used in the literature both for the whole fiber bundle (which contains all configurations), and for the base-space (which contains the shapes). We use the term in this latter meaning, and we reserve the name "Shape Bundle" for the former. Shape Spaces have been addressed in the literature of Statistics. At the same time, however, the concept of Shape Spaces is used in the exact same meaning in the literature of Geometric Mechanics, see for instance Littlejohn and Reinsch [69].

2.2 Observing Shape Spaces: the role of pose and motion

A *shape* has been defined in the previous section as the coordinates of a *configuration* of points relative to a very special *reference frame*, which has origin in the centroid of the points, the first axis aligned on the longer size of the configuration, which is defined to be the unit, and the second axis aligned on the longer size in the plane orthogonal to the first axis. Therefore, between a *shape* σ and a generic configuration $\bar{\mathbf{X}}$ there is just a *change of coordinates* $g_\gamma \in SE(3)$ and a scaling (if we neglect column defection):

$$\sigma = \frac{g_\gamma \bar{\mathbf{X}}}{\|g_\gamma \bar{\mathbf{X}}\|} \in \Sigma^N. \tag{2.16}$$

We call g_γ the *pose*, which is the change of coordinates between the shape's reference and the *world* reference frame and is described by six pose parameters $\gamma \in \mathbb{R}^6$ (three parameters for translation and three for rotation). We will use the name *structure* for the combination of shape and pose (i.e. a *configuration* of points).

Since the choice of the world reference is arbitrary, we can take it to be centered in the pupil of our eye, with the $Z-$axis (depth) aligned with the optical axis and the remaining ones ordered so as to form a right-handed orthonormal frame. Then the shape σ, seen from our eye, has a configuration

$$\bar{\mathbf{X}}_0 = g_\gamma^{-1}\sigma \in \mathbf{S}^{3N-1} \tag{2.17}$$

where $\bar{\mathbf{X}}_0$ has been adequately normalized. Now, if we start *moving* around with a (generalized) [2] velocity $v(t)\wedge \in se(3)$, our current reference changes relative to the world: in fact, at time t, the change of coordinates between our eye and the world is obtained by integrating the following equation

$$\dot{g}(\tau) = v(\tau) \wedge g(\tau) \in TSE(3) \tag{2.18}$$

from the initial time instant up to time t. $TSE(3)$ denotes the tangent bundle of the Euclidean group [3].

[2] The notation $se(3)$ stands for the Lie algebra of twists corresponding to rigid motions. See [81] for a brief account on $se(3)$ and its corresponding Lie group $SE(3)$.

[3] The tangent bundle of a manifold M is a special case of a fiber bundle, where the fibers are tangent planes T_pM at each point $p \in M$ of the manifold. Therefore, the tangent bundle is the collection of all possible tangent vectors at all possible points of a manifold: $TM = \cup_{p\in M} T_pM$. The projection of the fibers onto the base-space (the manifold itself in this case) is the trivial map $v_p \in T_pM \mapsto p \in M$.

The integral above cannot in general be written nor computed in closed-form unless velocity is constant, in which case $g(t) = e^{v\wedge t}$. For practical purposes, we take discrete samples of the time-interval $[0, t]$, and consider the corresponding discrete path $g(\tau_i), \tau_i \in [0, t], i = 1 \dots N_t$. The integral between time samples can be written as the exponential, under the assumption that velocity is piecewise constant. The integration of different samples is then just a composition (product) of rigid motions:

$$g(t) = \prod_{i=1}^{N_t} e^{v(\tau_i)\wedge(\tau_i - \tau_{i-1})} \in SE(3) \tag{2.19}$$

where it is intended that new factors are multiplied from the *left* [4]. We indicate this operation *symbolically* as

$$g(t) \doteq \int_0^t e^{v(\tau)\wedge\tau} d_{SE(3)}\tau. \tag{2.20}$$

The object, as seen from our current reference, has a configuration

$$\bar{\mathbf{X}}(t) = g(t)\bar{\mathbf{X}}_0. \tag{2.21}$$

What we can measure is the perspective projection of such a configuration onto the CCD surface, modeled as an image-plane. We write the perspective projection as

$$\begin{aligned} \pi : \mathbb{R}^{3\times N} &\longrightarrow (\mathbb{RP}^2)^N \\ \bar{\mathbf{X}} &\mapsto \pi(\bar{\mathbf{X}}) \end{aligned} \tag{2.22}$$

where each point $\mathbf{X}^i \in \mathbb{R}^3$ (a column of the configuration matrix), is transformed by

$$\pi(\mathbf{X}^i) = \frac{\mathbf{X}^i}{X_3^i} \in \mathbb{RP}^2. \tag{2.23}$$

[4] Often we take the sampling time as the unit of time, so that $\tau_i - \tau_{i-1} = 1$.

We use the same notation

$$\mathbf{y} = \pi(\mathbf{X}) = \begin{bmatrix} \frac{X_1}{X_3} \\ \\ \frac{X_2}{X_3} \\ \\ 1 \end{bmatrix} \in \mathbb{RP}^2 \tag{2.24}$$

depending upon the context, to indicate either the three *homogeneous coordinates*, or the two coordinates of a point on the image-plane (without a 1 appended)

$$\mathbf{y} = \pi(\mathbf{X}) = \begin{bmatrix} \frac{X_1}{X_3} \\ \\ \frac{X_2}{X_3} \end{bmatrix} \in \mathbb{R}^2. \tag{2.25}$$

Overall it looks like there are three reference frames involved in this picture: the *object* reference (with respect to which the scaled configuration is a *shape*), the *world* reference (which is arbitrary and fixed with respect to the object), and the *viewer* reference. We choose the world reference to coincide with the viewer's at time $t = 0$, so that the map between them is the unit rigid motion $e \in SE(3)$, which consists of the identity rotation matrix $R = I$ and zero translation $T = 0$. The corresponding changes of coordinates are what we call respectively *pose* (g_γ^{-1}: object to world) and *motion* ($g(t)$: world to viewer). We call *velocity* the instantaneous generalized velocity of the configuration relative to the viewer in the viewer's reference:

$$v(t)\wedge \doteq \dot{g}(t)g^{-1}(t) \in se(3). \tag{2.26}$$

In coordinates, v is represented by an instantaneous translational velocity V and an instantaneous rotational velocity ω: $v = (V, \omega) \in \mathbb{R}^6$. Overall we assume that we can measure

$$\bar{\mathbf{y}}(t) = \pi\left(g(t) \circ g_\gamma^{-1}\sigma\right) + n(t) \tag{2.27}$$

where $n(t) \in \mathcal{N}(0, \Sigma)$ is a white, zero-mean Gaussian process that models the noise in the localization of the projection on the sensor surface. Our goal here is to model the *shape* σ of the object being viewed, given

measurements on the retina (or image-plane) $\{\mathbf{y}^i(t)\}_{t\in[0,\tau]}\ \forall\ i = 1\dots N$. Such measurements, however, also depend upon other *unknown parameters*, namely pose and motion.

2.2.1 Models for observing shape, pose and motion

If we put together the measurement equation (2.27) and the dynamics of the parameters which appear in it, we end up with a model of the form

$$\begin{cases} \dot{\sigma} = 0 & \sigma(0) = \sigma_0 \in \Sigma^N \\ \dot{\gamma} = 0 & \gamma(0) = \gamma_0 \in \mathbb{R}^6 \\ \dot{g} = v \wedge g & g(0) = e \in SE(3) \\ \dot{v} \simeq 0 & v(0) = v_0 \in \mathbb{R}^6 \\ \bar{\mathbf{y}}(t) = \pi\left(g(t) \circ g_\gamma^{-1}\sigma\right) + n(t) \end{cases} \tag{2.28}$$

where we have assumed that the viewer moves with (approximately) constant velocity. We could now write an observer for the above model, which would ideally provide us with an estimate of shape, *along with an estimate of motion and pose.* We could as well pose the problem in a stochastic setting by modeling the constants as a Brownian motion and writing an observer in the form of an Extended Kalman Filter (EKF) [58, 17, 55] in the local coordinates of the state-space manifold. In such a case, however, the model for shape must take into account the fact that, even if the configuration is described by a Brownian motion in $\mathbb{R}^{3\times N}$, the shape coordinates *are not* a Brownian motion. The derivation of the correct distribution induced by projecting a Gaussian process in Euclidean space onto the Shape Space is derived in [77].

Regardless of the choice for a deterministic or a stochastic setting, for an observer to work properly, the model must be *observable*, which means that there must not be different initial conditions which give raise to different state trajectories that produce the *same measurements* for all times. We address this issue in section 4.1.

But, even without getting into the details of observability, one may notice that the initial conditions of the model (2.28) are certainly not observable. In fact, in the measurement equation the pose parameters g_γ and the motion parameters $g(t)$ always appear as a composition, so that if we substitute $\bar{g} \doteq \tilde{g}g(t)$ for $g(t)$, and $\bar{g}_\gamma \doteq \tilde{g}g_\gamma$ for g_γ we get the same measurement model for any choice of $\tilde{g} \in SE(3)$. Indeed, we also get

the same dynamics, since

$$\dot{\tilde{g}} = \frac{d}{dt}(\tilde{g}g) = \tilde{g}(v\wedge)g = \tilde{g}(v\wedge)\tilde{g}^{-1}\tilde{g}g = (\tilde{g}v)\wedge(\tilde{g}g) \doteq \bar{v}\wedge\bar{g}$$

and so for $\dot{\bar{v}} \doteq \tilde{g}\dot{v} = 0$. If we look more carefully into the model (2.28), however, we see that, while initial conditions for shape and pose are unknown, the initial condition for motion is known to be the identity transformation $e \in SE(3)$, since it expresses the fact that the viewer's reference coincides with the world reference at the initial time instant. This is a piece of information that must be taken into account. We do so in section 4.1, where we address the observability of the model (2.28), which we call *shape-pose-motion-velocity* model.

Before doing that, however, note that *pose* g_γ is "in between" motion and shape, and it could very well be removed by re-defining either one. For instance, we may choose $g(t)$ such that $g(0) = g_\gamma^{-1}$, which leads us to the model

$$\begin{cases} \dot{\sigma} = 0 & \sigma(0) = \sigma_0 \in \Sigma^N \\ \dot{g} = v\wedge g & g(0) = g_\gamma^{-1} \in SE(3) \\ \dot{v} \simeq 0 & v(0) = v_0 \in \mathbb{R}^6 \\ \bar{\mathbf{y}}(t) = \pi\left(g(t)\sigma\right) + n(t) \end{cases} \tag{2.29}$$

which we call *shape-motion-velocity* model. Both models we just described have the shortcoming of being ill-conditioned whenever the shape has any symmetry (see remark 2.1.1).

Alternatively, instead of characterizing shape relative to the object reference, we may choose the world as a reference, since the two are just related by a (static) change of coordinates. In such a case we get a model which involves *structure*, rather than shape:

$$\begin{cases} \dot{\bar{\mathbf{X}}}_0 = 0 & \bar{\mathbf{X}}_0(0) = \bar{\mathbf{X}}_0 \in \mathbf{S}^{3N-1} \\ \dot{g} = v\wedge g & g(0) = e \in SE(3) \\ \dot{v} \simeq 0 & v(0) = v_0 \in \mathbb{R}^6 \\ \bar{\mathbf{y}}(t) = \pi\left(g(t)\bar{\mathbf{X}}_0\right) + n(t). \end{cases} \tag{2.30}$$

This corresponds to choosing an alternative base of the Shape Bundle. Such a choice, of course, is not intrinsic to the object but, rather, it depends upon the particular position of the viewer relative to the

object at the initial time. However, once a particular configuration has been estimated (for instance that relative to the world), we can project it onto the Shape Space at any time. We call the last model the *structure-motion-velocity* model. We recall that we use the word *structure* for the combination of shape and pose.

To complete the picture, we note that also motion plays a fictitious role, since – its initial condition being known – it can be removed from the dynamic model simply by integrating the measurement equation. For instance we could transform (2.28) to

$$\begin{cases} \dot{\sigma} = 0 & \sigma(0) = \sigma_0 \in \Sigma^N \\ \dot{\gamma} = 0 & \gamma(0) = \gamma_0 \in \mathbb{R}^6 \\ \dot{v} \simeq 0 & v(0) = v_0 \in \mathbb{R}^6 \\ \bar{\mathbf{y}}(t) = \pi \left(\int_0^t e^{v \wedge \tau} d_{SE(3)} \tau \circ g_\gamma^{-1} \sigma \right) + n(t) \end{cases} \tag{2.31}$$

which we call the *shape-pose-velocity* model. Of course, we can also obtain a *configuration-velocity* model in the exact same way:

$$\begin{cases} \dot{\bar{\mathbf{X}}}_0 = 0 & \bar{\mathbf{X}}(0) = \bar{\mathbf{X}}_0 \in \mathbf{S}^{3N-1} \\ \dot{v} \simeq 0 & v(0) = v_0 \in \mathbb{R}^6 \\ \bar{\mathbf{y}}(t) = \pi \left(\int_0^t e^{v \wedge \tau} d_{SE(3)} \tau \bar{\mathbf{X}}_0 \right) + n(t). \end{cases} \tag{2.32}$$

It is also possible to operate a change of state coordinates from $\bar{\mathbf{X}}_0$ to $\begin{bmatrix} \bar{\mathbf{y}}_0 \\ \bar{Z}_0 \end{bmatrix}$, with $\bar{\mathbf{y}}_0 = \pi(\bar{\mathbf{X}}_0)$ and $\bar{Z}_0 = \bar{\mathbf{X}}_{0_3}$, so that the above model becomes

$$\begin{cases} \dot{\bar{\mathbf{y}}}_0 = 0 & \bar{\mathbf{y}}_0(0) = \bar{\mathbf{y}}_0 \in \mathbb{R}^{2 \times N} \\ \dot{\bar{Z}}_0 = 0 & \bar{Z}_0(0) = \bar{Z}_0 \in \mathbf{S}^{N-1} \\ \dot{v} \simeq 0 & v(0) = v_0 \in \mathbb{R}^6 \\ \bar{\mathbf{y}}(t) = \pi \left(\int_0^t e^{v \wedge \tau} d_{SE(3)} \tau \bar{\mathbf{y}}_0 \bar{Z}_0 \right) + n(t). \end{cases} \tag{2.33}$$

Although the above model is equivalent to (2.32), there is an advantage in such a choice of coordinates. In fact, if we describe our state $\mathbf{X}_0^i \in \mathbb{R}^3$ as a random walk, we need to attribute high variance to all components of $\mathbf{X}_0^i$. If, instead, we encode $\mathbf{X}_0^i$ with $(\mathbf{y}_0^i, Z_0^i)$, then we can lower the uncertainty on the initial estimates

for $\mathbf{y}_0^i$, for these are measured:

$$\mathbf{y}^i(0) = \mathbf{y}_0^i + n^i(0).$$

It is also possible to eliminate the states corresponding to $\mathbf{y}_0^i$ altogether:

$$\begin{cases} \dot{\bar{Z}}_0 = 0 & \bar{Z}_0 \in \mathbf{S}^{N-1} \\ \dot{v} = 0 & v(0) = v_0 \in \mathbb{R}^6 \\ \mathbf{y}(t) = \pi \left(\int_0^t e^{v \wedge \tau} d_{SE(3)} \tau \bar{\mathbf{y}}(0) \bar{Z}_0 \right) + n(t) + \beta_0 & \forall\, t\ > 0 \end{cases} \tag{2.34}$$

where the bias term β_0 in the measurement equation comes from having substituted $\bar{\mathbf{y}}(0) \doteq \bar{\mathbf{y}}_0 + n(0)$ for $\bar{\mathbf{y}}_0$. Such a bias is the price one pays for reducing the size of the state-space from $3N-1$ down to $N-1$. The above model, in particular, is substantially equivalent to the one introduced by Azarbayejani and Pentland in [5].

The models (2.28)–(2.34) are all different manifestations of the same underlying system. They are all *input/output equivalent*, in the sense that they have the same initial conditions and the same measured output. The models described can be obtained one from the other by simple changes of coordinates, integration or model reduction. This equivalence may surprise at first, since different models have different dimensions. However, in counting the dimensions one must include σ (3N-7 DOFs), γ (6 DOFs), g (6 DOFs), v (6 DOFs) as well as $g(0) = e$ (6 DOFs). We see immediately that all models have the same number of DOFs, namely 3N+5.

All above models are *integral* representations, in the sense that the state consists of the initial conditions, while integration is performed explicitly (although approximately, see eq. 2.19) in the measurement equation. For each model we may derive an equivalent *differential* representation, where integration is encoded in the state process. For instance, the structure-velocity model (2.32) is equivalent to

$$\begin{cases} \dot{\bar{\mathbf{X}}} = v \wedge \bar{\mathbf{X}} & \bar{\mathbf{X}}(0) = \bar{\mathbf{X}}_0 \in \mathbf{S}^{3N-1} \\ \dot{v} \simeq 0 & v(0) = v_0 \in \mathbb{R}^6 \\ \bar{\mathbf{y}}(t) = \pi \left(\bar{\mathbf{X}} \right) + n(t). \end{cases} \tag{2.35}$$

Similarly we may derive differential versions of all models described earlier. We may also sample such models to derive equivalent *discrete-time* ones.

Remark 2.2.1 *Since we claim that all models described so far are equivalent, one may wonder if different observers designed on different models would behave in the same way. Of course this is not the case. First of all, different models live in different nonlinear spaces. For instance (2.28) lives in the state space* $\Sigma^N \times SE(3) \times SE(3) \times se(3)$, *which has a highly non-trivial curvature structure and singularities in the local coordinate transformation. On the other hand, (2.32) lives on the space* $\mathbf{S}^{3N-1} \times \mathbb{R}^6$, *which has a simple and well-defined geometry. Furthermore, the normalization constraint can be enforced in the measurement constraint (see appendix 2.6), leaving us with an entirely linear state-space* $\mathbb{R}^{3N} \times \mathbb{R}^6$.

Remark 2.2.2 *If we wish to work in a stochastic setting, and use Brownian motions as models for the unknown parameters, we have to keep in mind that projections onto the Shape Space do not preserve the distributions of points on the fibers. Therefore, even if each point of a configuration is described as an independent Gaussian process, the shape coordinates are not independent nor Gaussian [77]. This makes the models involving shape coordinates impractical, in that a simple Extended Kalman Filter is not appropriate as a state estimator.*

Remark 2.2.3 *The models (2.28)–(2.34) are instances of "object-centered" models, while (2.35) is an instance of a "viewer-centered" model. As we have seen, the difference between such models is just an integral in the measurement equation.*

The main practical difference between these two classes is in the enforcing of the scale-factor ambiguity. In a viewer-centered frame, the scale factor needs to be updated at each step, thus generating a drift due to errors in the estimates of structure. Object-centered (or world-centered) models, on the other hand, can enforce the scale factor by imposing that any of the scaled state is (constant and equal to) a specified value.

2.3 Filtering Structure from Motion

In this section we discuss some qualitative issues related to the implementation of observers for estimating structure and/or motion from the models described in section 2.2.1. We have implemented observers in the form of traditional *Extended Kalman Filters*, assuming that the measurement noise is white, zero-mean, Gaussian and additive and that the unknown parameters are represented as first-order random walks.

Among all equivalent models described in section 2.2.1, we concentrate on the ones that describe *structure*,

rather than distinguishing between shape and pose. The reason for such a choice is two-fold. First, if we assume that each point of a configuration is described by an independent Gaussian process in Euclidean space, the shape coordinates are *not* Gaussian processes. Second, the local coordinatization of shape has singularities in the presence of symmetries.

We have implemented local EKFs for the *structure-motion-velocity* model (2.30), the *structure-velocity* model (2.32), the *reduced* depth-velocity model (2.34). In each instance we have enforced the normalization constraint either as a pseudo-measurement or as a model constraint by defining the configuration as points of a sphere or by saturating the filter along any state. The issue of scale normalization is addressed in appendix 2.6.

In this section we only discuss the qualitative properties of the models described in sections 2.2.1. A thorough simulation study of the performance of these methods in comparison with other schemes may be found in chapter 11.

Simulation setup

We have generated $N = 20$ points on a curved surface, placed $2m$ in front of the viewer. Such points are projected onto an ideal image-plane of 500×500 pixels, covering a visual angle of approximately 30^o. Gaussian noise is added to the image projections with a standard deviation of 5 pixels. The object rotates about its centroid by $5^o/frame$, so that the corresponding motion of the viewer is roto-translational and constrained onto the plane orthogonal to the rotation axis of the object. We have started each EKF with zero initial conditions.

Structure-motion-velocity model

The main feature of the structure-motion-velocity model (2.30) is the presence of an *on-line* integrator for motion. In order for motion to be estimated correctly, it is necessary to *enforce the initial condition* $g(0) = e$. This is done, in an EKF framework, by setting the variance of the initial condition for motion to zero. However, in order to avoid saturation, it is necessary to set the variance of the model error to a (small but) non-zero value, which causes the filter to have a small bias in the motion components (see figure 2.2). The initial conditions are zero for all states, except for the structure components, which are

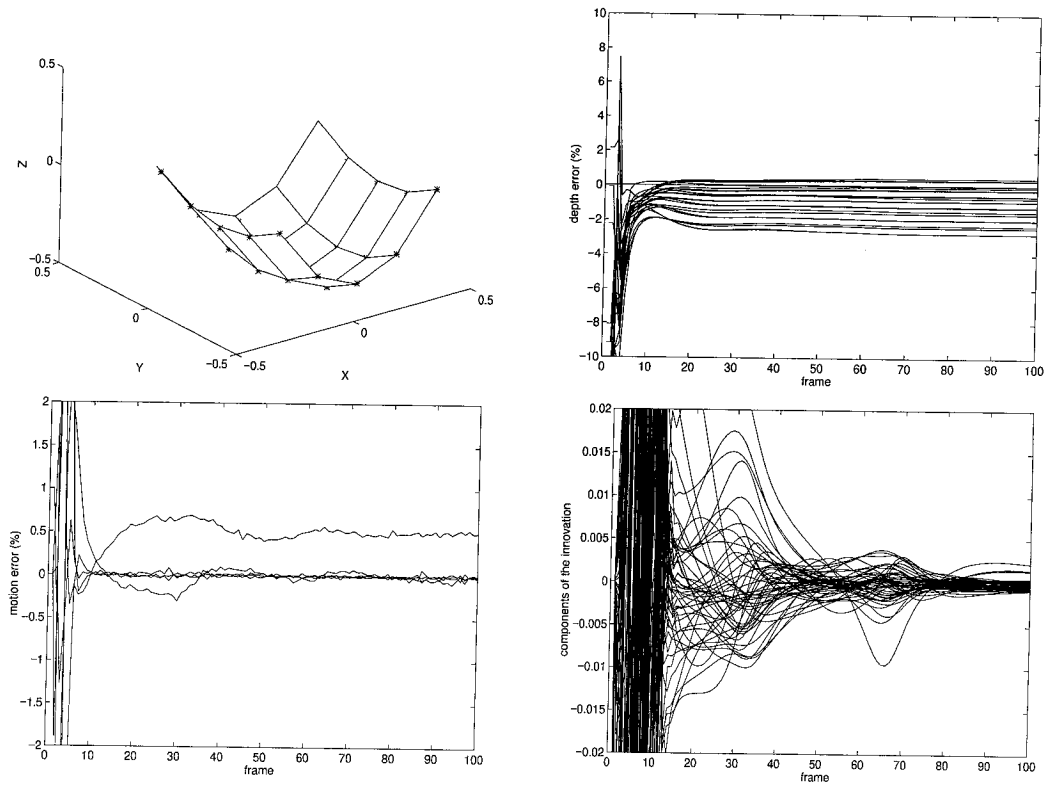

Figure 2.2: **Structure-motion-velocity model:** *the estimated position of each feature-point is shown in the top-left plot, along with the underlying true surface. The estimation error for the depth of each point is also show in the top-right plot, where it can be seen that there is a bias in the estimates (about 2 %). A bias can be also noticed in the estimates of motion in the lower-left plot. The residual error in the projection of each feature point (innovation) is reported in the lower-right plot. It can be seen that the components are quite correlated, and the decay is slow.*

initialized to their projective coordinates as measured at the initial time instant (as if all points were lying on the image-plane). The initial variance of the structure parameters, however, has been initialized to a high value ($10m$). It is possible to lower the initial variance on some of the parameters, as we describe next.

Structure-velocity model

In the structure-velocity model (2.32), the integral of the generalized velocity is performed *off-line*, so that there are no drifts due to the enforcement of the initial conditions for motion. We have also implemented an EKF based upon the alternative coordinatization of the model (2.33), which uses the image-plane coordinates and the depth of each point (rather than the 3-D coordinates of each point). Such a model has the advantage that the initial uncertainty on the states that correspond to (measured) image-plane coordinates can be set

equal to the variance of the measurement error, and therefore concentrate most of the initial uncertainty along the *depth* direction, rather than equally spreading it on the three coordinates of each point. We find that this is the implementation that achieves the best accuracy among the models implemented (see figure 2.3).

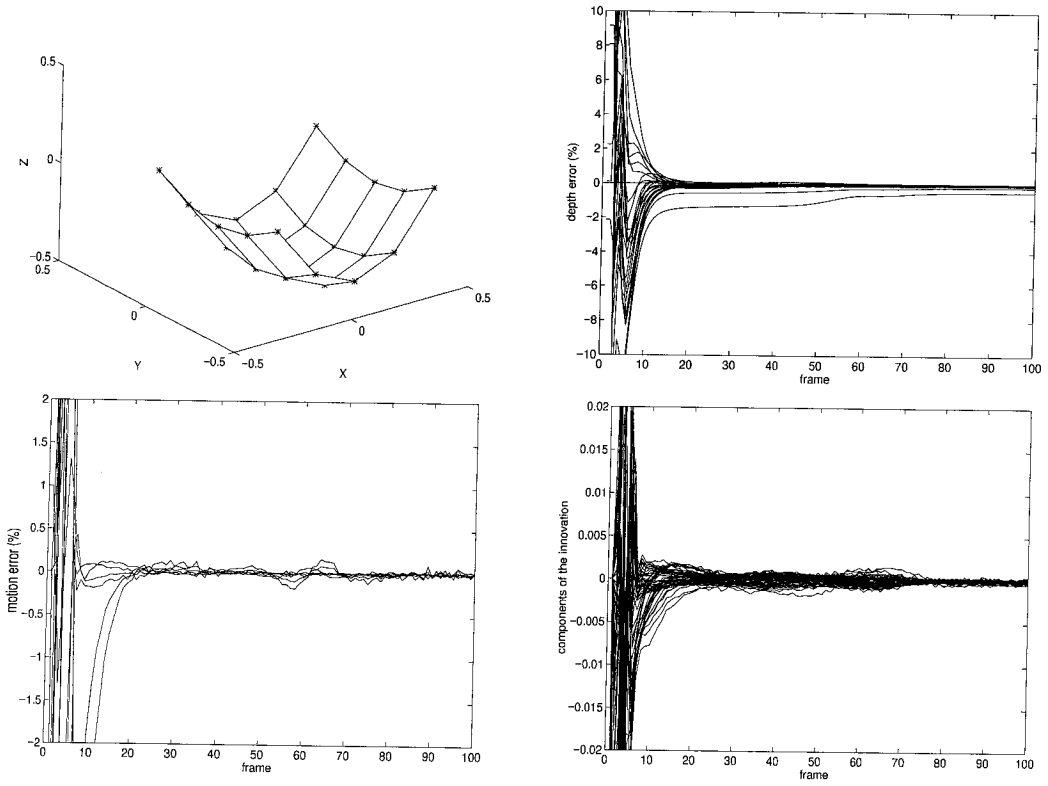

Figure 2.3: **Structure-velocity model:** *when motion is integrated off-line, we do not experience any bias in the estimates of structure, as it can be seen from the upper plots (left for the three-dimensional position of feature points and the truth three-dimensional surface, right for the estimation error in depth). The estimates of motion are also bias-free (lower-left), and the innovation is small and reasonably uncorrelated (lower-right).*

Reduced depth-velocity model

Similarly to [5], we have removed the states corresponding to the image-plane coordinates in the model (2.33), and implemented an EKF based upon the model (2.34). Such a model has the advantage of a smaller state-space, at the expense of a bias in the estimates due to the error in locating the features on the image-plane at the first time instant (figure 2.4). Such an error does not contribute to the innovation, and therefore it is never fed-back to the estimate of the state. The bias has already been reported in [5].

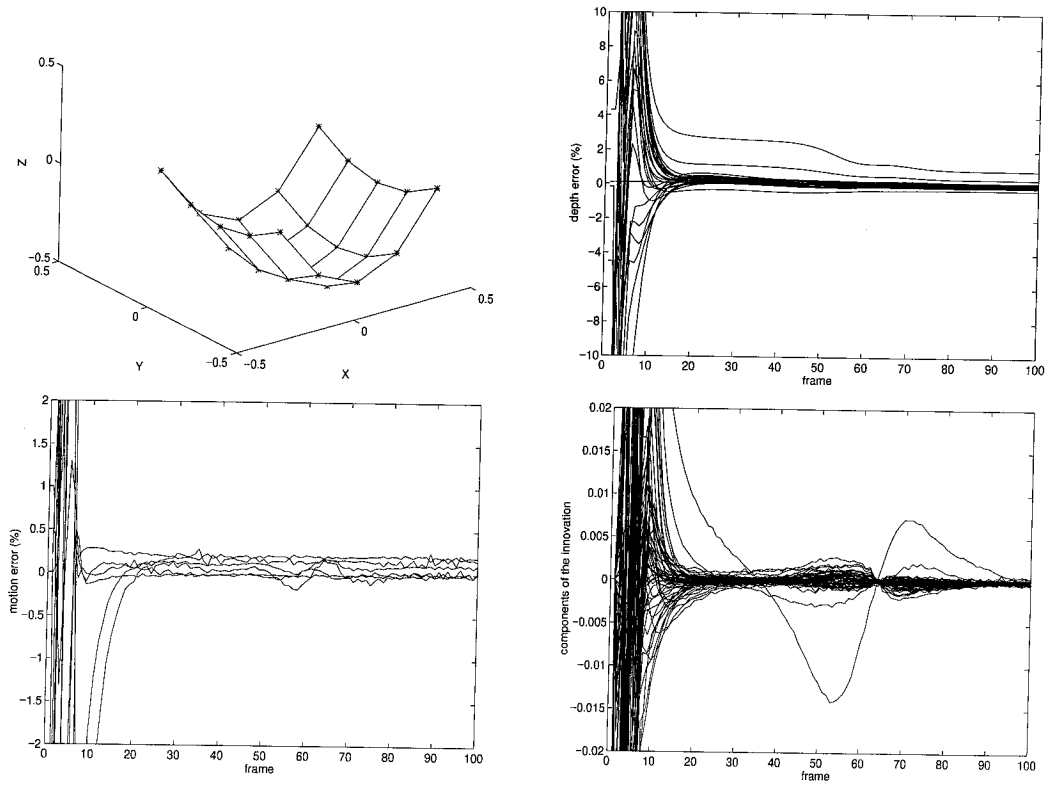

Figure 2.4: **Depth-velocity model:** *when we reduce the model by substituting the initial coordinates on the image-plane with their measured values, we introduce a small bias in the depth estimates (upper-right plot), as well as in the motion components (lower-left). The innovation is also more colored than before performing observer reduction.*

Instantaneous (differential) models

We have implemented an EKF based upon the instantaneous model (2.35). Such a filter suffers a significant drift in the estimates, which is due to the fact that the *scale factor* ambiguity must be propagated across time (since the states corresponding to structure are relative to the viewer). This causes an accumulation of errors that quickly causes the estimator to diverge in the experimental conditions described in this section.

A qualitative study on real images

In this section we report the results of experiments on a sequence of images kindly provided by AIACE (the International Association of Computational Archeology). The experiment consists in the application of the structure-velocity model to a sequence of images of an archeological site in Marzabotto–Italy. We have first selected and tracked automatically a number of features using a multi-scale version of the so-called SSD

algorithm (see appendix A for details), and then estimated the three-dimensional coordinates of each feature point. Since there is no ground truth to compare with, we can only observe that the qualitative properties of the structure are estimated correctly (for instance the square angle between the faces of the wall, see figure 2.5).

The bottom line is that the noise-level achieved by standard feature-tracking technique is easily handled by simple EKFs based upon the models described in this paper. Extensive simulation studies which test the performance of each scheme in the presence of varying noise level, aperture angle, number of visible features, sampling period, initial conditions, bas-relief ambiguity etc. are reported in chapter 11.

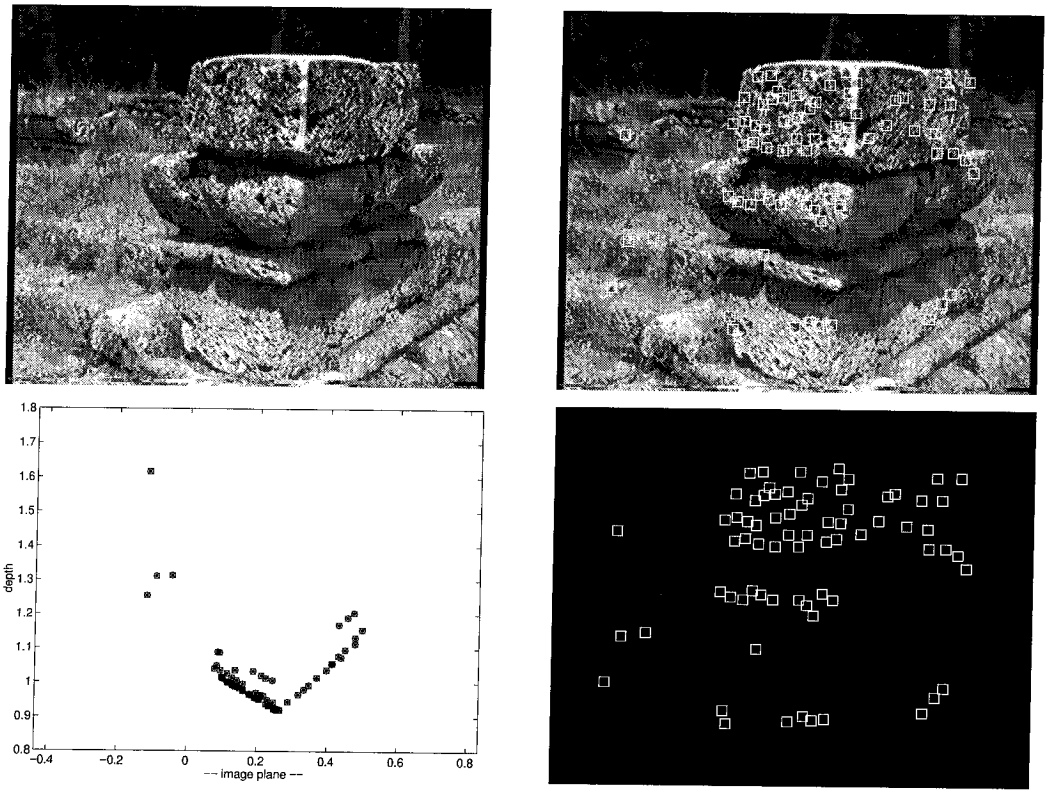

Figure 2.5: *(Top-left) one of 45 images of an archeological site in Marzabotto–Italy. (Top-right) feature-points are automatically selected based upon local gradient criteria and showed as the area enclosed in a white box. (Bottom-right) the tracking of features is used as input for the structure-velocity model, which estimates the normalized position in 3-D of each feature point. A top view of the estimated structure is shown in (bottom-left).*

2.4 Model reduction and invariance

There are still some aspects of the models described in section 2.2.1 which have not been elucidated: we know that, if an object is visible, it must be in front of the observer, i.e. $Z^i \geq 0 \, \forall \, i$. Moreover, no points are allowed to lie on the focal plane $Z = 0$. These are constraints that add to the normalization constraints, described in appendix 2.6, which are needed in order to reduce the set of indistinguishable states (see chapter 4). This may be done at the price of transforming the state from the linear state-space $\mathbb{R}^9$ to the differentiable manifold with boundary $\mathbb{R}^2 \times \mathbb{R}^+ \times \mathbf{S}^2 \times SO(3)$, and $\mathbf{S}^2$ is the two-sphere [11]). However, an appropriate model should include such constraints *explicitly* into the state representation.

In the previous sections we have seen some cases in which it is possible to *eliminate* some parameters from the state of the model without affecting its overall functioning. For instance, in transforming the model (2.30) onto (2.32), we have exploited the fact that we know the initial condition to perform the integration in closed-form. The model (2.34), on the other hand, is obtained by *eliminating the states that can be measured directly*, in a fashion similar to the technique of *observer reduction* [57].

It is possible to push the idea of observer reduction in order to eliminate uninteresting states at each level of derivative in the observability co-distribution, as we will see in chapter 6. The price we pay is that at each level of reduction we introduce a derivative of the measurements, and a bias term if they are noisy. The advantages depend upon the applications, and in certain cases they can be far more than computational.

As an example, consider the models described in section 2.2.1. It can be noticed that there are *no modeling errors* (at least in the case of constant velocity), since the dynamics consist of purely geometric constraints. Now suppose that we want to *fit a model* to the visible points in 3-D. For instance, instead of representing shape as a collection of points, we want to represent it as the surface – chosen within a class of parametric models – that best interpolates the configuration. The residual of such an interpolation is now a *modeling error*. Such a modeling error, however, should only affect the *structure parameters*, not the motion parameters. Therefore, in order for the structure modeling error not to affect the estimate of motion, it is necessary to *decouple* their dynamics in the observer. A way to ensure that modeling errors in structure do not affect the estimates of motion is to use the dynamics of motion in order to eliminate it from the state model. Modeling shape as a surface model is beyond the scope of this paper; however, in section 2.4.1 we derive a model for point-wise structure which is independent of motion parameters. Then, in principle, one

could choose a parametric model class for the structure parameters and substitute the depth of each point in the model with the surface parameters.

Modeling structure as a surface is also an effective way of handling *occlusions*. In fact, occluded feature-points can be removed and new features added to the measurement model without changing the dimension of the state (which consists of surface parameters), and without having to initialize structure parameters. This can also be achieved by eliminating structure parameters altogether, so that the reduced model is also independent of the particular choice of the representation of structure. In section 2.4.2 we motivate this approach, which we then discuss in more detail in following chapters.

2.4.1 Motion-independent structure estimation

Consider the *structure-velocity* model in its instantaneous version (2.35), with the change of state coordinates $\mathbf{X}^i \longrightarrow (\mathbf{y}^i, Z^i)$:

$$\begin{cases} \dot{\mathbf{y}}^i = \mathcal{C}(\mathbf{y}^i, Z^i) \begin{bmatrix} V \\ \omega \end{bmatrix} \\ \dot{Z}^i = \mathcal{F}(\mathbf{y}^i, Z^i) \begin{bmatrix} V \\ \omega \end{bmatrix} \quad \forall\, i = 1 \dots N \\ \dot{V} = 0 \\ \dot{\omega} = 0 \\ \tilde{\mathbf{y}}^i(t) = \mathbf{y}^i(t) + n^i(t) \end{cases} \tag{2.36}$$

where we have defined

$$\mathcal{C}(\mathbf{y}^i, Z^i) \doteq \begin{bmatrix} \frac{\mathcal{A}^i}{Z^i} & \mathcal{B}^i \end{bmatrix} \tag{2.37}$$

$$\mathcal{F}(\mathbf{y}^i, Z^i) \doteq \begin{bmatrix} 0 & 0 & 1 & -\mathbf{y}_2^i Z^i & \mathbf{y}_1^i Z^i & 0 \end{bmatrix}. \tag{2.38}$$

If we stack a number of vectors $\mathbf{y}^i$ on top of each other and we neglect the effects of the measurement errors, then we can assume that we can measure directly the first $2N$ states and therefore eliminate them. We may

also eliminate V and ω simply by solving the state equation of $\mathbf{y}^i$ and substituting in the dynamics of Z^i:

$$\begin{cases} \dot{\bar{Z}} = \mathcal{F}(\bar{\mathbf{y}}, \bar{Z})\mathcal{C}^\dagger(\bar{\mathbf{y}}, \bar{Z})\dot{\bar{\mathbf{y}}} & \bar{Z} \in \mathbf{S}^{N-1} \\ \mathcal{C}^\perp(\bar{\mathbf{y}}, \bar{Z})\dot{\bar{\mathbf{y}}} = 0. \end{cases} \tag{2.39}$$

The symbol $\dagger$ stands for the pseudo-inverse, and $\perp$ for the orthogonal complement; $\bar{Z}$ and $\bar{y}$ stand for the vectors that collect all Z^i, y^i. Note that, in the above model, both $\bar{\mathbf{y}}$ and $\dot{\bar{\mathbf{y}}}$ play the role of measurements.

Remark 2.4.1 *Note that in the above model we do not need to make the assumption of constant-velocity (or small acceleration), for velocity has been eliminated from the model. The drawback is that the measurement noise now also affects the state model in a non-additive, non-linear fashion, as it can be seen by substituting* $\mathbf{y}$ *with* $\mathbf{y} + n$.

As a matter of motivation, suppose we want to model structure as a parametric surface, described by

$$\begin{array}{rcl} \mathcal{S} : W \times U & \longrightarrow & \mathbb{R}^3 \\ (\alpha, u) & \mapsto & S_\alpha(u) = \begin{bmatrix} u \\ \bar{Z}(\alpha) \end{bmatrix} \end{array} \tag{2.40}$$

where $W \subset \mathbb{R}^a$ and $U \subset \mathbb{R}^2$. We can choose the image plane as a local parametrization of the surface, so that $u = \mathbf{y}$. If we now distinguish – among the parameters α – the ones that influence the *pose* of the surface (call them γ) from the ones that influences its *shape* (call them σ), we can write a model that is formally identical to (2.28), where now $\gamma \in \mathbb{R}^6$ and $\sigma \in \mathbb{R}^s$, and the measurement equation is substituted by

$$\bar{\mathbf{y}}(t) = \pi \left(g(t) \circ g_\gamma^{-1} S_{\sigma,\gamma}(\bar{\mathbf{y}}_0) \right) + n(t). \tag{2.41}$$

We may employ the same techniques used in this section in order to to derive a model similar to (2.39), which only involves surface parameters and not motion nor velocity. In the case of a surface there are non-trivial issues related to the choice of the model and its validation, which add to the problem of designing an observer that exploits the non-linear and non-additive structure of the noise. In this thesis we concentrate on the simplest "point-wise" representation of structure and, therefore, we do not address these issues.

2.4.2 Towards structure-independent motion estimation

In the dynamical models described in the previous sections we assume to measure the trajectories of the output over an extended period of time:

$$\{\mathbf{y}^i(t) \; \forall \; i = 1 \dots N \; , \; t \in [t_0, t_1]\}. \tag{2.42}$$

Such an assumption is equivalent to assuming that we have solved the *correspondence problem*, which is that we know which point on the image-plane corresponds to which across time.

Even in the discretized models, it is usually reasonable to assume that we can solve the correspondence problem for a certain length of time. It is, however, extremely difficult to maintain a "label" of each point for a long time. Also, some of the features may disappear, because they exit the field of view, or because they become occluded.

It is fairly simple to handle the appearance of new features: suppose that, at some instant of time τ, we have a new measurement point $\mathbf{y}^i$ that enters the field of view, and we want to include it in the measurement set. In order to do that, it is necessary to project such a point onto the slice of the configuration space that corresponds to the viewer's frame at $t = 0$. If we have a current estimate for the motion g,

$$g_\tau \doteq \hat{g}(\tau) \tag{2.43}$$

as well as the projection of the point in question at the same time

$$\mathbf{y}^i_\tau \doteq \mathbf{y}^i(\tau) \tag{2.44}$$

then it is immediate to see that

$$\mathbf{y}^i(t) = \pi(g(t) \circ g_\gamma^{-1} \circ g_\tau^{-1} \mathbf{y}^i_\tau Z^i_\tau). \tag{2.45}$$

We may include g_τ^{-1}, as well as g_γ^{-1}, into g and therefore fall in the cases described in section 2.2.1.

Remark 2.4.2 *If we follow the above procedure, every time a new feature enters the model and its depth is initialized, the initialization error affects also motion parameters, for they are coupled through the state*

model. Therefore, even in the presence of a smooth motion, the estimates we get are discontinuous due to the initialization errors in the structure parameters.

Furthermore, in the presence of occlusions we need to remove states from the dynamical model, which also affects the estimates of other states.

A way to overcome the problem of occlusion is to exploit the invariance of structure to *eliminate it* from the model, so as to have a structure-independent model for estimating motion. This is the principal argument of coming chapters, and we anticipate some of the main issues in the next section.

2.5 Decoupling and reduction as a modeling strategy

When facing a high-dimensional optimization problem, it is important to unravel the geometry of space of unknown parameters, in order to see whether there are "slices" where the parameters evolve independently in the cost objective. This responds to the need of decomposing a high-dimensional optimization task into the solution of a number of smaller, simpler and better conditioned problems.

In the case of structure and motion estimation, the work of Longuet-Higgins [73] pioneered this approach, by decoupling structure from the motion parameters, which he encoded in a 3×3 matrix, called *essential matrix*. Adiv [3] and Heeger and Jepson [45] further decoupled the translational velocity from the rotational velocity.

We will re-derive the constraints of Heeger/Jepson and Longuet-Higgins within a unified procedure. We will start from the dynamical models described in section 2.2.1, and construct the so-called *reduced-order observer* [57] both for the continuous-time and the discrete-time models. These result, respectively, in the so-called "Subspace constraint" and the "Epipolar (or Coplanarity) constraint", now interpreted as nonlinear implicit models of a special class (so-called Exterior Differential Systems [16]) with parameters on a manifold (chapters 3 and 5). Such a manifold is a 5-dimensional space, called Essential manifold, in the discrete-time case of Longuet-Higgins and the 2-dimensional sphere in the continuous-time case of Heeger andJepson.

This asymmetry between continuous and discrete time, which cannot be resolved in the context of the reduced-order observer, is what will motivate us towards alternative strategies for reducing the model, which we discuss in chapter 7.

2.5.1 The basic reduced-order observer: simultaneous depth and motion estimation

The reduced-order observer [57] is a long-established technique for reducing the dimension of an observer for a dynamical system. The basic idea consists in "solving" the measurement equation for some of the states, and then substitute into the model equation. The states that have been eliminated are no longer part of the state-space, and their state equation becomes a new measurement equation, which involves derivatives of the measurements. The original measurement equation becomes now trivial, for it has been used to define the states to be eliminated.

For instance, consider the simple linear model

$$\begin{cases} \dot{x}_1 = a_{11}x_1 + a_{12}x_2 \\ \dot{x}_2 = a_{21}x_1 + a_{22}x_2 \\ y = c_1x_1 + c_2x_2 \end{cases} \tag{2.46}$$

and "solve" the measurement equation for x_2, so that $x_2 \doteq \frac{y-c_1x_1}{c_2}$. If we now substitute x_2 into the dynamic equations, we get a new state model for x_1 which does not involve x_2 but has an "output injection" term, and a constraint involving the measurements y and $\dot{y}$ and the unknown state x_1:

$$\begin{cases} \dot{x}_1 = (a_{11} - a_{12}\frac{c_1}{c_2})x_1 + \frac{a_{12}}{c_2}y \\ \frac{1}{c_2}\dot{y} + (a_{22}\frac{1}{c_2} - a_{12}\frac{c_1}{c_2^2})y = (a_{11}\frac{c_1}{c_2} - a_{22}\frac{c_1}{c_2} + a_{12}\frac{c_1^2}{c_2^2} + a_{21})x_1. \end{cases} \tag{2.47}$$

The original measurement equation is now the identity $y = y$. We may re-write the above model as

$$\begin{cases} \dot{x}_1 = \tilde{a}x_1 + ky \\ \tilde{y} = \tilde{c}x_1 \end{cases} \tag{2.48}$$

where $\tilde{y}$ hides a time-derivative of the measured output y. It is possible to get rid of this undesirable effect by either an output-dependent change of coordinates, as done in the original reduced-order observer [57], or by integrating the measurement equation over a sample time interval.

This simple idea, applied to the structure-velocity model (2.32) produces exactly the depth-velocity model (2.34), which we had derived from heuristic arguments.

2.5.2 Pushing observer reduction: structure-independent motion estimation

Although the depth-velocity model has fewer states than the structure-velocity model, it still involves structure parameters and, therefore, it can vary in time due to occlusions and appearance of new features. The next step consists in applying the same idea of the reduced-order observer to the already-reduced model in order to get rid of structure parameters altogether.

Continuous-time: the Subspace model

If we apply the idea of the reduced-order observer twice to the structure-velocity model, in the first run we can eliminate $2N$ states, corresponding to the measured projections of each feature-point, and be left with $N+6$ states, describing the depth of each point and the motion parameters. In the second run we can "solve" the new measurement equation, which in fact corresponds to the image motion field (and is approximated by the optical flow), for the depth parameters.

Since the expression of the image motion field $\dot{\mathbf{x}}$ is linear both in the inverse depth and the rotational velocity, one may eliminate both depth and rotation, as done in Heeger and Jepson [45].

In chapter 5 we will view the resulting constraint as a reduced dynamical model, which happens to be in the form of a so-called "Exterior Differential Systems" [16]. The only unknown in such a model is the direction of translation, which is modeled as a point on a sphere.

Discrete-time: the Essential model

The idea of the reduced-order observer may be applied also to the discrete-time version of the structure-velocity model. The tool to be used for eliminating the depth parameters is the so-called "Epipolar geometry", which essentially resorts to the well-known coplanarity constraint, first derived by Longuet-Higgins [73].

In chapter 3 we will interpret such a constraint asa discrete dynamical system with unknown parameters on the so-called "Essential manifold". Such parameters encode the relative orientation of the camera frame between successive time instants.

2.5.3 Asymmetry between continuous and discrete-time

In the continuous-time case we will push the idea of the reduced-order observer up to the point in which we have a model with only two parameters (the spherical coordinates of the direction of translation). This will not work in the discrete-time case. In fact, the rotation parameters appear through the exponential map, which we cannot invert in closed-form in order to substitute it into the model equation and apply the tools of the reduced-order observer.

Therefore, there is an asymmetry between the instantaneous case and the discrete-time case. This will motivate us to explore alternative methods for reducing the state of the observer.

2.5.4 "Explicit" versus "implicit" decoupling

Although it is not always possible to decouple the unknown parameters in closed-form, it is possible to do so implicitly by imposing that some function of the parameters is held constant. We will see how this leads to a reduction of the model by constraining it onto a subspace of the parameter space. For instance, we may impose that the image of a point, a line, or a plane remains fixed. This procedure identifies slices of the parameter manifold where the model is constrained to evolve. For instance, these manifolds are 4 and 3-dimensional submanifolds of the Essential manifold, when a point or a line are fixated, and the 2-dimensional sphere (also a submanifold of the Essential manifold), in the case in which a plane is fixated. Thus, we may interpret the compensation of the motion of a point, a line, or a plane, as a geometric stratification of the Essential manifold. By restricting the model to the appropriate slices, we derive 4, 3 and 2-dimensional dynamic constraints, the latter being the discrete-time equivalent of the Heeger and Jepson's constraint.

Appendix

2.6 Scale factor normalization

Let us consider the structure-velocity model in its reduced version, where only the *depth* of each point at the initial instant is encoded and noise is being neglected:

$$\begin{cases} \dot{\bar{Z}}_0 = 0 \ \in \mathbb{R}^N \\ \dot{v} = 0 \ \in se(3) \\ \bar{\mathbf{y}} = \pi \left(g(t) \bar{\mathbf{y}}(0) \bar{Z}_0 \right). \end{cases} \tag{2.49}$$

As we have anticipated in previous sections, the above model is *not observable*, for there is an overall scalar ambiguity affecting the depth of each point and the translational component of motion. Therefore, an additional scale constraint must be imposed. Such a scale constraint can be imposed either to the *pose* of the configuration, by enforcing that some point (or any combination of the points) is at some prescribed depth, or to the *shape* of the configuration, by enforcing the set of dots to have a prescribed size.

2.6.1 Normalization of pose

Suppose we wish to get rid of the scale-factor ambiguity by imposing that some particular point has a specified depth, for instance

$$Z^1(t) = \rho \ \forall \ t. \tag{2.50}$$

There are essentially two ways to impose such a constraint: one is to enforce it in the state model of the filter, the other is to add a measurement constraint.

Pose normalization in the state model

The first option, which has been chosen by Azarbayejani et al. [5], consists in generating an initial condition for the chosen point, for instance

$$Z^1(0) = \rho \tag{2.51}$$

with zero-variance, $\Sigma_{Z^1} = 0$, so that the filter is saturated along one direction, and its states evolve along

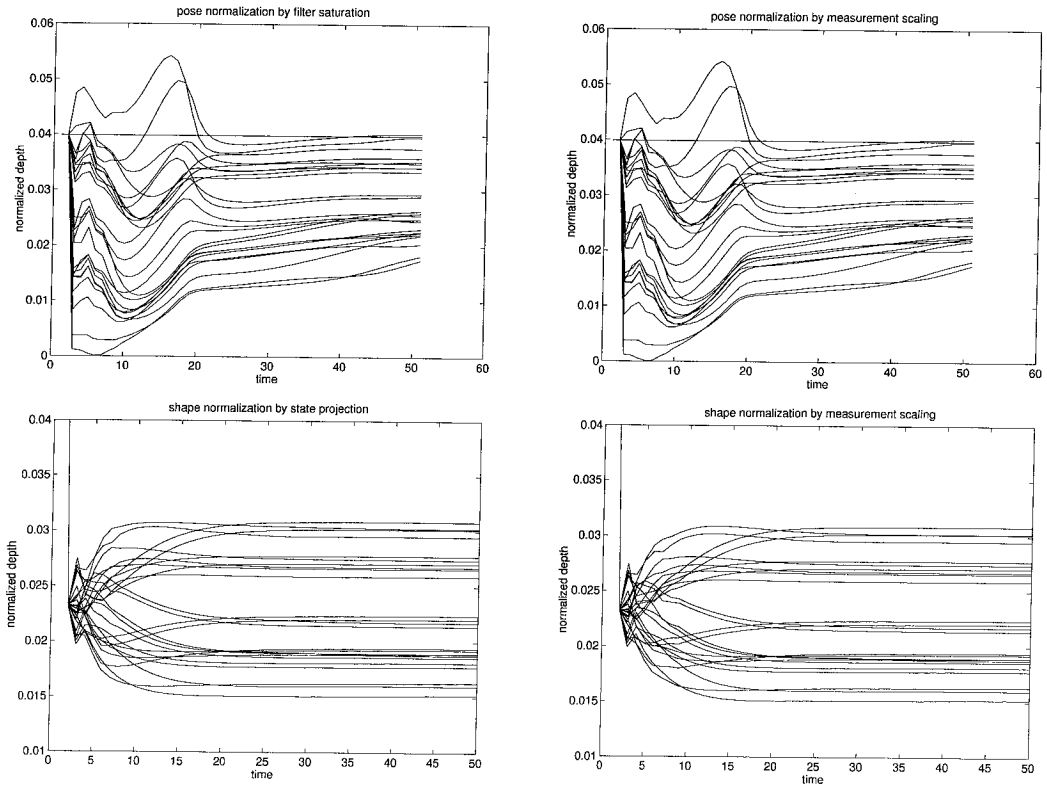

Figure 2.6: **Scale normalization:** *pose can be normalized either by saturating a state (upper-left) or by adding a measurement constraint (upper-right). Alternatively, we may constrain the overall size of the points to be scaled, by either forcing the state of the model onto a sphere (lower-left) or by adding a measurement constraint (lower-right). The normalized estimates of the depth of each point, as reported in the above plot, indicate that normalization of shape helps achieving faster convergence and smoother estimates.*

the orthogonal subspace. In essence one works with an observer for an un-observable model, where one row of the gain matrix is kept at zero.

Pose normalization in the measurement model

Alternatively, we may leave the state model untouched (without imposing saturation), and add a measurement constraint of the form

$$\rho = Z^1(t) \tag{2.52}$$

with variance of the measurement error $\Sigma_{Z^1} = 0$. This setup is very similar to the previous one, except that the filter now operates in normal conditions and its model is made observable by a *dummy* measurement equation.

2.6.2 Normalization of shape

As an alternative to normalizing *pose*, we may impose a constraint on the overall size of the configuration, for instance by imposing

$$\|\bar{Z}(t)\| = \rho \; \forall \; t. \tag{2.53}$$

Again, such a constraint can be imposed on the state model

$$\dot{\bar{Z}}_0 = 0 \; \in T\mathbf{S}^{N-1} \tag{2.54}$$

or we could add the measurement equation

$$\rho = \|\bar{Z}(t)\|. \tag{2.55}$$

Such a normalization works on average better than the normalization of pose, for all states (and therefore all measurements) contribute to the constraint, as shown in figure 2.6.

Shape normalization in the state model

In order to impose the constraint $\bar{Z}_0 \in \mathbf{S}^{N-1}$, the filter state should evolve on $\mathbf{S}^{N-1}$, and therefore have $N-1$ independent states. In order to do this, we could devise a system of local coordinates. However, these would have singularities and it would require multiple charts.

A method to circumvent this problem consists in embedding the sphere in $\mathbb{R}^N$ and then project the update at each time onto the sphere. Such a solution is certainly not fundamental, but effectively the most practical, since there is no need to add new dummy measurements, or set artificially some variances to zero.

The correct approach would be to formulate the filter directly on the sphere, without using any coordinatization. This intrinsic representation is beyond the scope of this thesis, and we will therefore not address it here.

Shape normalization in the measurement model

This case works exactly like the one in section 2.6.1, except that the new dummy measurement constraint is now

$$\rho = \|\bar{Z}(t)\|. \tag{2.56}$$

2.7 Inner products and Riemannian metrics on the Shape Space

We have defined a Shape Space through a series of transformations of the *total* space of configurations $\mathbb{R}^{3\times N}$:

$$\begin{array}{ccccccccc} \mathbb{R}^{3\times N} & \stackrel{\text{centering}}{\longrightarrow} & <\mathbf{1}>^{\perp} & \stackrel{\text{defection}}{\longrightarrow} & \mathbb{R}^{3\times(N-1)} & \stackrel{\text{scaling}}{\longrightarrow} & \mathbf{S}^{3(N-1)-1} & \stackrel{\text{mod-out}}{\longrightarrow} & \mathbf{S}^{3(N-1)-1}/SO(3) \\ \bar{\mathbf{X}} & & \bar{\mathbf{X}}_c & & \bar{\mathbf{X}}_{co} & & \bar{\mathbf{X}}_{cos} & & \sigma. \end{array} \tag{2.57}$$

In doing so we have implicitly identified the two spaces

$$\mathbb{R}^{3\times N}/\{SE(3)\times\mathbb{R}\} \Longleftrightarrow \mathbf{S}^{3(N-1)-1}/SO(3). \tag{2.58}$$

It is indeed immediate to see that two configurations $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ belong to the same orbit under $SE(3)\times\mathbb{R}$ (rigid motion and scale) if and only if $\bar{\mathbf{X}}_{cos}$ and $\bar{\mathbf{Y}}_{cos}$ belong to the same orbit in $SO(3)$. We may define a map from $\mathbf{S}^{3(N-1)-1}/SO(3)$ to $\mathbb{R}^{3\times(N-1)}/SO(3)$ by taking $\bar{\mathbf{X}}_{cos}\in\mathbf{S}^{3(N-1)-1}$ and considering it as an element of $\mathbb{R}^{3\times(N-1)}$. Such a map, restricted to the equivalence class, can be proven to be an *isometric embedding* [19]. Therefore, given two shapes, their distance is equal to the distance of the corresponding scaled-centered shapes, considered as elements of $\mathbf{S}^{3(N-1)-1}/SO(3)$ [19].

We are now interested in deriving a *distance* between two shapes on Σ^N, exploiting the fact that we can consider them as elements of $\mathbb{R}^{3\times N}/SO(3)$.

Let us consider a configuration, which is a generic element of $\mathbb{R}^{3\times N}$. Such a space can be identified with the space $\mathcal{H}(3,N)$ of linear maps between $\mathbb{R}^N$ and $\mathbb{R}^3$, which is a Hilbert space with the inner product

$$\begin{array}{rcl} <\cdot,\cdot>_{\mathcal{H}}\colon \mathcal{H}(3,N)\times\mathcal{H}(3,N) & \longrightarrow & \mathbb{R} \\ (\bar{\mathbf{X}},\bar{\mathbf{Y}}) & \mapsto & <\bar{\mathbf{X}},\bar{\mathbf{Y}}>_{\mathcal{H}}=\text{trace}(\bar{\mathbf{X}}^T\bar{\mathbf{Y}}). \end{array} \tag{2.59}$$

To be more precise, the above map should be defined between $T\mathcal{H}(3,N)\times T\mathcal{H}(3,N)$ and $\mathbb{R}$ but, $\mathcal{H}(3,N)$ being a linear space, its tangent bundle [5] coincides with the space itself: $T\mathcal{H}(3,N)=\mathcal{H}(3,N)$. Such an inner product induces a *Riemannian metric*, which can be used to define a *norm* $\|\bar{\mathbf{X}}\|^2_{\mathcal{H}}\doteq<\bar{\mathbf{X}},\bar{\mathbf{X}}>_{\mathcal{H}}\ \forall\,\bar{\mathbf{X}}\in$

[5] The tangend bundle to a manifold M, indicated by TM, is the collection of all tangent planes at all possible points $p\in M$ of the manifolds: $TM\doteq\cup_{p\in M}T_pM$.

$T\mathcal{H}(3,N)$, and a *global distance* on $\mathcal{H}(3,N)$ simply by $d_{\mathcal{H}}(\bar{\mathbf{X}},\bar{\mathbf{Y}})^2 \doteq < \bar{\mathbf{X}}-\bar{\mathbf{Y}}, \bar{\mathbf{X}}-\bar{\mathbf{Y}} >_{\mathcal{H}}$. The way this distance is obtained is by integrating the Riemannian metric along the shortest path between $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ (called a *geodesic*), which in the simple case above is a straight line.

This does not work for a *curved manifold* such as the Shape Space. What we have to do in such a case is to define a Riemannian metric, and then integrate it along a geodesic in order to define the distance between two shapes. D. G. Kendall has defined a metric on the Shape Space, which is calls *procrustean metric*, which defines a distance

$$d_{\Sigma}(\bar{\mathbf{X}},\bar{\mathbf{Y}}) \doteq \inf_{R\in SO(3)} \|\bar{\mathbf{X}}_{cos} - R\bar{\mathbf{Y}}_{cos}\|_{\mathcal{H}}. \tag{2.60}$$

It has been shown [19] that such a metric is indeed the one that is derived from the Riemannian metric, and a closed-form expression is given by

$$d_{\Sigma}(\bar{\mathbf{X}},\bar{\mathbf{Y}})^2 = \|\bar{\mathbf{X}}\|^2 + \|\bar{\mathbf{Y}}\|^2 - 2\text{trace}(\bar{\mathbf{X}}\bar{\mathbf{Y}}^T). \tag{2.61}$$

Note that this distance is different from the distance of the two points *considered as points on the total space* (the Shape Bundle). In fact, if considered arbitrary points in $\mathbb{R}^{3\times(N-1)}$, the distance between $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ induced by the inner product would be $\|\bar{\mathbf{X}}\|^2 + \|\bar{\mathbf{Y}}\|^2 - 2\text{trace}(\bar{\mathbf{X}}^T\bar{\mathbf{Y}})$.

Chapter 3 Observer reduction in the discrete-time case: motion estimation on the essential manifold

In previous chapters we have seen how the problem of estimating structure and motion from a sequence of images can be formalized in terms of state estimation for certain nonlinear dynamical systems. Since we have adopted a point-wise representation for structure, occlusions are a problem as they affect the size of the state-space and cause discontinuities in the estimates.

In this chapter we show how the invariance of shape can be used in order to *eliminate structure altogether* from the state of our dynamical model, in the same spirit of the reduced-order observer, ending up with a model for motion independent of structure.

Background and notation for the chapter

In this chapter we are going to describe a representation of rigid motion via the so-called Essential matrices. Such matrices are denoted with $\mathbf{Q}$. We will use the properties of rotation matrices $R \in SO(3)$ and those of skew-symmetric matrices. Such matrices can be always written in the form of a matrix (operator) $(T\wedge)$, which acts on three-dimensional vectors $\mathbf{X} \in \mathbb{R}^3$ according to the rule

$$(T\wedge)\mathbf{X} \doteq T \wedge \mathbf{X} \tag{3.1}$$

where $\wedge$ is the usual cross-product. an explicit expression for $(T\wedge)$ is given by

$$T\wedge = \begin{bmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{bmatrix}. \tag{3.2}$$

More details can be found in [81]. We will also mention tangent bundles, since Essential matrices can be naturally defined as elements of the tangent bundle of the rotation group $SO(3)$. As in previous chapters, we call $\mathbf{x} = \pi(\mathbf{X})$ the projection of a feature point of coordinates $\mathbf{X} \in \mathbb{R}^3$ onto the image-plane, modeled as the real projective plane $\mathbb{R}\mathrm{P}^2$. We indicate the coordinates of $\mathbf{x}$ as $\mathbf{x} = [x\ y\ 1]^T$.

Outline of the chapter

We start by defining the space of Essential matrices. These can be interpreted as a "concise" way to represent rigid motions using a single 3×3 matrix. Then we show how such a representation plays a central role in the problem of estimating motion, since it allows deriving constraints on the motion parameters which are independent of structure. Such constraints are well-known in the vision literature and methods for exploiting them in order to estimate motion have been proposed both in closed-form and iteratively from two views (stereo). We propose to

view such constraints as (nonlinear and implicit) dynamical models with parameters on the manifold of Essential matrices, and outline a method for identifying these parameters.

3.1 The Essential manifold

A rigid motion may be represented as a point in the Lie group $SE(3)$, which can be embedded in the linear space $\mathcal{GL}(4)$ (and hence exploit the matrix product as composition rule) and is in local correspondence with $\mathbb{R}^6$ via the exponential coordinates and the isomorphism between $so(3)$ and $\mathbb{R}^3$, as in [81]. We now discuss an alternative matrix representation of rigid motion which is derived from the so-called "Essential matrices" introduced by Longuet-Higgins [73].

Consider a rigid motion $g = (T, R) \in SE(3)$; then $T\wedge \in so(3)$ is a skew-symmetric matrix. We define the space of *"Essential matrices"* as

$$E \doteq \{SR \mid \ R \in SO(3)\ ,\ S = (T\wedge) \in so(3)\} \subset \mathbb{R}^{3\times 3}. \tag{3.3}$$

Clearly the Essential space does not inherit the group structure from the general linear group $\mathcal{GL}(3)$, since $\mathbf{Q}_1, \mathbf{Q}_2 \in E$ does not imply $\mathbf{Q}_1 + \mathbf{Q}_2 \in E$. One possible way of imposing the group structure is by forcing a group morphism with $SE(3)$, for which it is necessary to "unfold" T, R from $\mathbf{Q} = (T\wedge)R \in E$, perform the group operation on $SE(3)$ and then collapse the result into E. We will see later in this section a way of unfolding an Essential matrix into its rotation and translation components.

3.1.1 Properties of the Essential manifold

The Essential space has many interesting geometrical properties: it is an algebraic variety that can be defined as the subset of the 3×3 matrices that satisfy the following polynomial equations:

$$\mathbf{Q} \in E \subset \mathbb{R}^{3\times 3} \Leftrightarrow \begin{cases} \det(\mathbf{Q}) = 0 \\ \frac{1}{2}\mathrm{tr}(\mathbf{Q}\mathbf{Q}^T)\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T\mathbf{Q}. \end{cases} \tag{3.4}$$

This characterization leads to algebraic methods for estimating Essential matrices from pairs of images. The interested reader may consult [30, 79]. We do not follow this "algebraic" approach here. Rather, we consider E to be a topological manifold, for which we will give an explicit local-coordinate characterization. The

reasons for our choice is that finding roots to polynomial equations makes it difficult to retain a geometric interpretation. Furthermore, since roots are found in the complex field, the resulting variety turns out to have singularities [79]. On the other hand, using a differential geometric approach, we can easily identify E with the tangent bundle to the rotation group, defined as $TSO(3) \doteq \cup_{R \in SO(3)} T_R SO(3)$, which is a smooth manifold and therefore has no singularities.

In fact, elements of $TSO(3)$ are all and only the 3×3 matrices obtained by taking a tangent vector to the origin of the rotation group, $S \in T_e SO(3) = so(3)$, which are elements of the Lie algebra of skew-symmetric 3×3 matrices, and pushing it forward by (right) multiplication to a point $R \in SO(3)$ of the rotation group:

$$S = T\wedge \in so(3) = T_e SO(3) \longrightarrow SR \in T_R SO(3). \tag{3.5}$$

Therefore, the tangent bundle to the rotation group is the set of matrices which are the product of skew-symmetric matrices and rotation matrices, which is exactly the way we have defined the Essential manifold.

The following claim, due to Huang and Faugeras and reported by Maybank [79], gives a simple characterizing property of the space of Essential matrices.

Claim 3.1.1 *(Huang and Faugeras, 1989)*

Let $\mathbf{Q} = \mathbf{U}\Sigma\mathbf{V}^T$ *be the Singular Value Decomposition (SVD) [39] of a matrix in* $\mathbb{R}^{3\times 3}$. *Then*

$$\mathbf{Q} \in E \Leftrightarrow \Sigma = \Sigma_0 = diag\{\lambda\ \lambda\ 0\} \mid \lambda \in \mathbb{R}^+.$$

Proof:

($\Rightarrow$) let $\mathbf{Q} = SR \mid R \in SO(3), S \in so(3)$; $\sigma(\mathbf{Q})$, the set of singular values of $\mathbf{Q}$, is such that $\sigma(\mathbf{Q}) = \sqrt{\sigma(\mathbf{Q}\mathbf{Q}^T)}$. Next observe that $\mathbf{Q}\mathbf{Q}^T = SS^T = -S^2$. Also $\forall\ S \in so(3)\ \exists\ !\ T \in \mathbb{R}^3 \mid S = (T\wedge)$, and the singular values of S^2 are $\{\|T\|^2, \|T\|^2, 0\}$. Hence if $\mathbf{Q} \in E$, it has two equal singular values and a zero singular value.

($\Leftarrow$) let $\mathbf{Q} = \mathbf{U}\Sigma_0\mathbf{V}^T$ be a Singular Value Decomposition. Let furthermore $R_Z(\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ +1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ be a rotation of $\frac{\pi}{2}$ about the axis $[0\ 0\ 1]^T$, then

$$\mathbf{Q} = \mathbf{U}\Sigma_0\mathbf{V}^T = \mathbf{U}\Sigma_0 R_Z^T(\pm\frac{\pi}{2})\mathbf{U}^T\mathbf{U}R_Z(\pm\frac{\pi}{2})\mathbf{V}^T.$$

Now call $R \doteq \mathbf{U}R_Z^T(\pm\frac{\pi}{2})\mathbf{V}^T$ and $S \doteq \mathbf{U}\Sigma_0 R_Z(\pm\frac{\pi}{2})\mathbf{U}^T$; it is immediate to see that $RR^T = R^TR = I$ and $S^T = -S$. From the uniqueness of the SVD, it follows that this decomposition is unique, modulo the sign in $R_Z(\pm\frac{\pi}{2})$. **Q.E.D.**

Remark 3.1.1 *Note that, since* $\mathbf{Q} \doteq \mathbf{U}\Sigma\mathbf{V}^T \in E \Leftrightarrow \Sigma = \text{diag}\{\lambda\ \lambda\ 0\}$, *there is one degree of freedom in defining the basis components of the subspaces* $< \mathbf{V}_{.3} >^{\perp}$ *and* $< \mathbf{U}_{.3} >^{\perp}$, *which corresponds to rotating the orthogonal bases* $< \mathbf{V}_{.1}, \mathbf{V}_{.2} >$ *and* $< \mathbf{U}_{.1}, \mathbf{U}_{.2} >$ *about their orthogonal complements. However, the effects cancel out in the multiplications when defining* R *and* S *as in the proof above.*

3.1.2 Local coordinates of the Essential manifold

For any given rigid motion $(T, R) \in SE(3)$, there exists an Essential matrix $\mathbf{Q}$ defined by $\mathbf{Q} \doteq (T\wedge)R$. We are interested now in the inverse problem: *given an Essential matrix* $\mathbf{Q}$, *can we extract its rotational and translational components? Is the correspondence* $\mathbf{Q} \leftrightarrow (T, R)$ *unique?*

Consider the following map, defined locally between E and $\mathbb{R}^6$

$$\begin{aligned} \Phi : E &\rightarrow \mathbb{R}^3 \times SO(3) \rightarrow \mathbb{R}^3 \times \mathbb{R}^3 \\ \mathbf{Q} &\mapsto \begin{bmatrix} \pm\|\mathbf{Q}\|\mathbf{U}_{.3} \\ \mathbf{U}R_Z(\pm\frac{\pi}{2})\mathbf{V}^T \end{bmatrix} = \begin{bmatrix} T \\ e^{\Omega\wedge} \end{bmatrix} \mapsto \begin{bmatrix} T \\ \Omega \end{bmatrix} \end{aligned} \tag{3.6}$$

where $\mathbf{U}, \mathbf{V}$ are defined by the Singular Value Decomposition (SVD) [39] of $\mathbf{Q} = \mathbf{U}\Sigma\mathbf{V}^T$; $\mathbf{U}_{.3}$ denotes the third column of $\mathbf{U}$ and $R_Z(\frac{\pi}{2})$ is a rotation of $\frac{\pi}{2}$ about the axis $[0\ 0\ 1]^T$. Note that the map Φ defines the local coordinates of the Essential manifold modulo two signs; therefore, the map Φ associates to each element of the Essential space four distinct points in local coordinates. This ambiguity may be resolved

in the context of the visual motion estimation problem by imposing the *"positive depth constraint"*, which means that each visible point lies in front of the viewer. In a case like this we will be able to identify a unique local coordinates homeomorphism, as discussed in section 3.3.1. The inverse map of Φ is simply

$$\begin{array}{rcl} \Phi^{-1} : \mathbb{R}^3 \times \mathbb{R}^3 & \rightarrow & E \\ \begin{bmatrix} T \\ \Omega \end{bmatrix} & \mapsto & (T\wedge)e^{(\Omega\wedge)} \end{array}$$

which is smooth.

3.1.3 Projection onto the Essential manifold

The claim 3.1.1 suggests a simple "projection" of a generic 3×3 matrix onto the Essential manifold:

$$\begin{array}{rcl} pr_{<E>} : \mathbb{R}^{3\times 3} & \rightarrow & E \\ M & \mapsto & \mathbf{U}\ \mathrm{diag}\{\lambda, \lambda, 0\}\ \mathbf{V}^T \end{array} \tag{3.7}$$

where $\mathbf{U}, \mathbf{V}$ are defined by the SVD of $M = \mathbf{U}\ \mathrm{diag}\{\sigma_1, \sigma_2, \sigma_3\}\ \mathbf{V}^T$, and $\lambda \doteq \frac{\sigma_1+\sigma_2}{2}$. It follows from the properties of the SVD [39] that $pr_{<E>}(M)$ minimizes the Frobenius distance of M from the Essential manifold [44, 79].

3.2 Role of the Essential manifold in Structure from Motion

When a rigid object is moving between two time instants t and $t+1$, the coordinates $\mathbf{X}$ of a point at time t, their correspondent $\mathbf{X}'$ at time $t+1$ [1], and the translation vector T are coplanar (fig. 3.1). Their triple product is therefore zero. This is true of course also for $\mathbf{x}$, $\mathbf{x}'$ and T, since $\mathbf{x}$ is the projective coordinate of $\mathbf{X}$ and therefore the two identify the same direction in $\mathbb{R}^3$, interpreted as the "ray-space" model of $\mathbb{R}P^2$ [86]. If we call $(T, R) \in SE(3)$ the rigid motion between the camera reference at time t and the one at time $t+1$, so that $\mathbf{X}' = R\mathbf{X} + T$, then we can write the triple product in a common reference frame, for instance the

[1] In this section we use $\mathbf{X}$ for $\mathbf{X}(t)$ and $\mathbf{X}'$ for $\mathbf{X}(t+1)$ for simplicity of notation.

camera's at time $t+1$, as

$$\mathbf{X'}^T T \wedge R\mathbf{X} = \mathbf{x'}^T T \wedge R\mathbf{x} = 0. \tag{3.8}$$

Let us define $\mathbf{Q} \doteq (T\wedge)R \in E$. The above coplanarity constraint, which is known as the "Essential constraint" or the "epipolar constraint", can be written for each visible point as

$$\mathbf{x}^i(t+1)^T \mathbf{Q} \mathbf{x}^i(t) = 0 \qquad \forall i = 1 \dots N. \tag{3.9}$$

Estimating motion then corresponds to identifying the model

$$\begin{cases} (\mathbf{Q}\mathbf{x}^i(t))^T \mathbf{x}^i(t+1) = 0 & \mathbf{Q} \in E \\ \mathbf{y}^i = \mathbf{x}^i + n^i & \forall i = 1 \dots N, \quad n^i \in \mathcal{N}(0, \Sigma_{n^i}) \end{cases} \tag{3.10}$$

which we call the *Essential model.* Since the Essential model is linear in $\mathbf{Q}$, we use the improper notation

$$\chi(t+1)\mathbf{Q}(t) \doteq \chi_{\mathbf{x'}(t),\mathbf{x}(t)}\mathbf{Q}(t) = 0 \qquad \chi \in \mathbb{R}^{N\times 9}$$

where χ is an $N \times 9$ matrix combining $\mathbf{x}^i, \mathbf{x'}^i$ and $\mathbf{Q}$ is interpreted as a nine-dimensional vector obtained by stacking the columns of the 3×3 matrix $\mathbf{Q}$ on top of each other. In the following we will not distinguish between $\mathbf{Q}$ interpreted as a matrix in $\mathbb{R}^{3\times 3}$ and a nine-dimensional column vector. The generic row of χ has the form $[xx',\ yx',\ x',\ xy',\ yy',\ y',\ x,\ y,\ 1\ ]$. We will use the notation $\chi(t)$ when emphasizing the time-dependence, while we will write $\chi_{\mathbf{x'}(t),\mathbf{x}(t)}$ when highlighting which vectors are used for constructing χ.

3.2.1 Two-views closed-form solutions: Longuet-Higgins revisited

Suppose we are given two views of N points, where $N \geq 8$. Then it is possible to write 8 or more constraints in the form

$$\chi\mathbf{Q} = 0. \tag{3.11}$$

These are *linear* constraints on the elements of a *generic (unstructured) vector* $\mathbf{Q}$ that solves the above equation. We may therefore use standard least-squares to estimate a *generic* vector $\tilde{\mathbf{Q}}$, for instance using

the Singular Value Decomposition:

$$\chi = U\Sigma W^T \longrightarrow \tilde{\mathbf{Q}} = W_{.9}. \tag{3.12}$$

The vector $\tilde{\mathbf{Q}}$ is the null-space of the matrix χ. In general, due to noise, χ will be full-rank, so one can just choose the singular vector $W_{.9}$ corresponding to the smallest singular value in order to obtain the best two-norm approximation of the null-space of χ. The 3×3 matrix $\tilde{\mathbf{Q}}$ is then obtained by re-ordering the elements of the nine-dimensional vector $W_{.9}$ into a matrix.

Since we have not used the fact that $\mathbf{Q}$ must belong to the Essential manifold, it is quite clear that, in general, $\tilde{\mathbf{Q}} \notin E$. In order to fix this problem, we may project $\tilde{\mathbf{Q}}$ down to the Essential manifold:

$$\hat{\mathbf{Q}} = \mathrm{pr}_{<E>}(\tilde{\mathbf{Q}}). \tag{3.13}$$

Once this is done, the motion parameters are obtained just as the local coordinates of the matrix $\hat{\mathbf{Q}}$ as in equation (3.6).

The procedure just outlined is Essentially equivalent to the scheme originally proposed by Longuet-Higgins [73].

Remark 3.2.1 *The method proposed by Longuet-Higgins consists in separating the (nonlinear) problem of estimating motion parameters into two combined linear tasks. The resulting solution is not optimal in any sense, since the constraints on the parameters of the Essential manifold are not enforced during the estimation step but, rather, generic unstructured parameters are first estimated, and then their structure is imposed a-posteriori.*

3.2.2 Two-views iterative solutions: Horn's Relative Orientation

Instead of decomposing the nonlinear task of finding motion parameters into two combined linear problems as proposed by Longuet-Higgins, one could try to solve directly for the motion parameters by minimizing some norm of the Essential constraint

$$(T, \Omega) = \arg\min_{T,\Omega} \|\mathbf{x}^{i\prime} T \wedge e^{\Omega\wedge}\mathbf{x}^i\| \tag{3.14}$$

with some choice of norm. This, in general, cannot be done in closed-form. Horn [51] proposed to use a gradient-descent algorithm to solve iteratively for the motion parameters.

This procedure has the advantage of enforcing all constraints on the parameters [2]. However, since we are using an iterative procedure, we have to deal with the inevitable presence of local minima which arises by using a general-purpose iteration that is not aware of the geometry of the underlying problem.

3.3 Dynamic solution: the "Essential filter"

The Essential constraint (3.9) has been used over the past decade in a variety of methods for estimating relative orientation from two views. Here we propose an alternative way of looking at the problem: rather than considering the coplanarity constraint as a set of algebraic equations on the motion parameters given two images, we view it as a *dynamical model.* Such a dynamical model is in a rather peculiar form, which is that of a linear and implicit system, and has unknown parameters that are elements of a topological manifold. Estimating motion amounts to performing the identification of the Essential model, where the parameters are constrained on the Essential manifold.

Since the Essential constraint is an homogeneous equation, and hence defined only up to a scale factor, we may restrict $\mathbf{Q}$ to belong to $\mathbf{S}^8$ instead of $\mathbb{R}^9$. It is customary to set the norm of translation to be unitary; this can be done without loss of generality, as long as translation is not zero. The zero-norm translation case can be dealt with separately, and we discuss it in section 3.3.3. For simplicity we now assume $\|\mathbf{Q}\|_2 = \|T\| = 1$. At each time instant we have a set of N constraints in the form

$$\chi_{\mathbf{x}'(t),\mathbf{x}(t)}\mathbf{Q}(t) = 0,$$

therefore, $\mathbf{Q}$ lies at the intersection between the Essential manifold and the linear variety $\chi^{-1}_{\mathbf{x}'(t),\mathbf{x}(t)}(0)$ (see fig. 3.2).

Note that, even after imposing unit norm, there is still a sign indeterminacy in $\mathbf{Q}$, which accounts for the two possible solutions $\mathbf{Q}_1 = +\mathbf{Q}$ and $\mathbf{Q}_2 = -\mathbf{Q}$ of the Essential constraint. These become four after being transformed to local coordinates. This ambiguity can be overcome by imposing the positive depth constraint

[2] Horn used unit quaternions as an embedded (non-minimal) representation of rotations.

as it will be done in section 3.3.1.

As time progresses, the point $\mathbf{Q}(t)$, corresponding to the actual motion, describes a trajectory on E (and a corresponding one in local coordinates) according to

$$\mathbf{Q}(t+1) \doteq \mathbf{Q}(t) + n_{\mathbf{Q}}(t).$$

The last equation is indeed just a *definition* of the right-hand side, as we do not know $n_{\mathbf{Q}}(t)$. The identity of $n_{\mathbf{Q}}(t)$ and the meaning of the sign + in the above equation will be unraveled in section 3.4.2. For now, we will consider the previous equation to be a discrete-time dynamical model for $\mathbf{Q}$ on the Essential manifold, with $n_{\mathbf{Q}}$ as *unknown* input. In the case of constant-velocity motion we have $n_{\mathbf{Q}} = 0$. If we accompany the above equation with the Essential constraint, we get

$$\begin{cases} \mathbf{Q}(t+1) \doteq \mathbf{Q}(t) + n_{\mathbf{Q}}(t) & \mathbf{Q} \in E \\ 0 = \chi_{\mathbf{x}'(t),\mathbf{x}(t)}\mathbf{Q}(t) & \\ \mathbf{y}^i = \mathbf{x}^i + n^i & \forall\, i = 1 \ldots N. \end{cases} \tag{3.15}$$

Now the visual motion estimation problem is characterized as the estimation of the state of the above model, which is defined on the Essential manifold. It can be seen that the system is "linear" (both the state equation and the Essential constraint are linear in $\mathbf{Q}$). E, however, is not a linear space. We will see how to solve the estimation task in section 3.4.

The observability/identifiability of the Essential models is addressed in chapter 4. It is proven that the model is globally observable under general position conditions. Such conditions are satisfied if the viewer's path and the visible objects cannot be embedded in a quadric surface of $\mathbb{R}^3$.

3.3.1 Choosing the local coordinates for the Essential manifold

The map Φ introduced in eq. (3.6) defines the local coordinates of the Essential space modulo a sign in the direction of translation and in the rotation angle of R_Z. Therefore, the map Φ associates to each element of the Essential space 4 distinct points in local coordinates. This ambiguity can be resolved by imposing the *"positive depth constraint"*, i.e. that each visible point lies in front of the viewer [34, 44, 73, 74, 109]. Consider one of the four local counterparts of $\mathbf{Q} \in E$, and the triangulation function $d_{\mathbf{x},\mathbf{x}'} : E \to \mathbb{R}^{1+1}$,

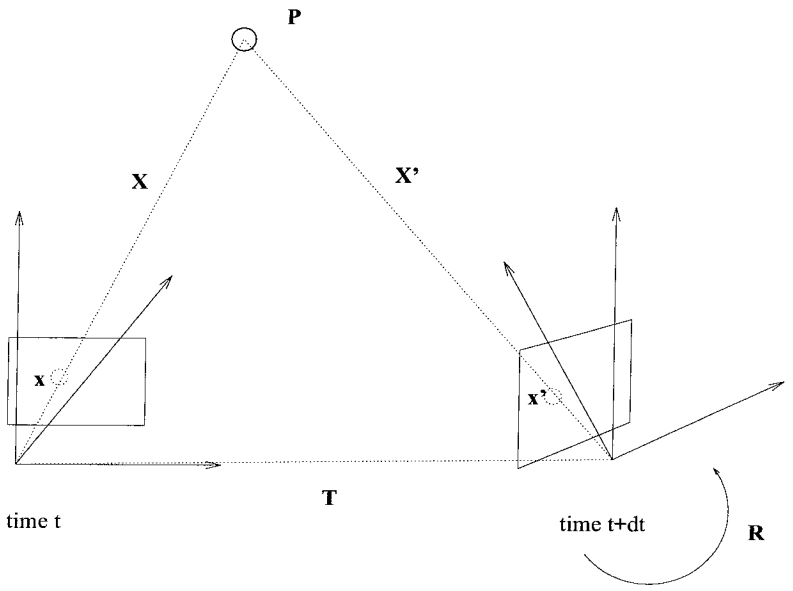

Figure 3.1: *The coplanarity constraint*

with $d_{\mathbf{x},\mathbf{x}'}(\mathbf{Q}) = [Z, Z']^T$, which gives the depth of each point as a function of the projection and the motion parameters (it is just the intersection of corresponding projection rays, see figure 3.1). Note that it is locally smooth away from zero translation. Therefore, given any N point-matches with projective coordinates $\mathbf{x}^i, \mathbf{x}'^i$ we may use Φ as a local coordinate chart for the following set, which we call the *"normalized Essential manifold"*:

$$\begin{aligned}\mathbf{E} &\doteq E \cap d^{-1}_{\mathbf{x},\mathbf{x}'}(\mathbb{R}^2_+)^N \cap \mathbf{S}^8 = \\ &= \{\mathbf{Q} = SR | R \in SO(3), S \doteq T\wedge \in so(3), \|T\| = 1, d_{\mathbf{x}^i,\mathbf{x}'^i}(\mathbf{Q}) > 0 \ \forall i = 1..N\} \end{aligned} \tag{3.16}$$

where $\mathbb{R}_+$ is the positive open half space of $\mathbb{R}$, and $d^{-1}_{\mathbf{x},\mathbf{x}'}$ denotes the preimage of $d_{\mathbf{x},\mathbf{x}'}$. Consider Φ restricted to $\mathbf{E}$. It follows from the properties of the SVD that Φ is continuous and, furthermore, bijective. The normalized Essential manifold thus defined is a topological manifold of dimension 5, since we have imposed the metric constraint $\|T\| = 1$.

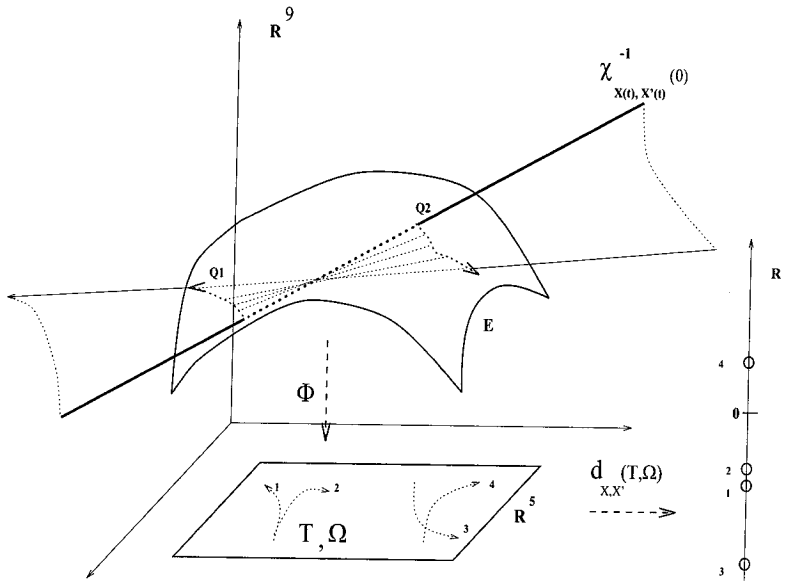

Figure 3.2: *Structure of the motion problem on the Essential space.*

3.3.2 Propagating scale information

It is well known that from visual information it is only possible to recover the structure and the motion modulo a scale factor multiplying the translational velocity and the depth of the visible points (see chapter 4). Such a scale ambiguity is captured by the homogeneous nature of the Essential constraint (3.9). However, as soon as we are given some scaling information about the scene *at one time instant*, we can rescale the scene and the estimated velocity to its appropriate values.

Suppose we are given the distance between two visible "reference" points in space $\|\mathbf{X}_{r1} - \mathbf{X}_{r2}\| = \rho$. Once motion has been estimated, with a normalized translational velocity, it can be used to estimate the "normalized structure" $\tilde{\mathbf{X}}^i$ via triangulation [93]. By matching the distance between the reference points in the normalized structure with its reference value, we can rescale both the depth of each point and the direction of translation simply by $\|\tilde{\mathbf{X}}_{r1} - \tilde{\mathbf{X}}_{r2}\| = \rho\|T\|$.

3.3.3 Dealing with zero-translation

So far we have assumed that $\|T\| \neq 0$, and we have defined the normalized Essential manifold based upon the constraint $\|T\| = 1$. It is easy to see that the condition $\|T\| = 0$ defines a "thin-set" in the parameter space. Due to the noise in the measurements, there is always a translation which is least-squares compatible

with the observations. However, one may ask what happens when the system is close to such a configuration. When translation is almost zero, there is little parallax in the projected coordinates of the visible objects, which makes the estimates of the depth and those of the direction of translation ill-conditioned.

Luckily enough, we do not need to worry about the structure of the scene, since it does not enter our dynamic model, and about the direction of translation, since its estimate will be weighted by the scale, which is $\|T\| \cong 0$. However, we would still like to estimate the correct rotational velocity. Here the definition of the normalized Essential manifold comes at hand. In fact, the estimation scheme will estimate some direction of translation $\hat{T}$ such that $\|\hat{T}\| = 1$ regardless the scale of T, so that the correct rotational component of the local coordinates can be computed. In the experimental section we will show an experiment in which the system crosses a region in the parameter space where $T = 0$ and $\Omega \neq 0$.

Remark 3.3.1 *The Essential constraint (3.9) defines a unique Essential matrix (up to scale) only if 8 or more point matches are given. If 5 or more matches are available, one may extract directly the motion parameters from the Essential constraint (up to a finite number of solutions). Early motion estimation schemes from two frames, based upon the Essential matrices, needed at least 5 or 8 point matches in order to estimate motion [51, 73, 54]. However, since the Essential model is recursive and integrates motion over time, it does not need to have a minimum number of features visible at each time instant, as long as the observability conditions are satisfied (see chapter 4). Therefore, using a filter based upon the Essential model (3.10) allows us to maintain the motion estimates even when crossing regions of the ambient space with less than 5 visible features.*

3.4 Solving the estimation task

At this point we are ready to address the problem of recursively estimating motion from an image sequence. There are two approaches that may be derived naturally from the Essential model.

The first approach we describe consists of composing the equations (3.15) with the local coordinate chart Φ, ending up with a *nonlinear* dynamical model for motion in $\mathbb{R}^5$. At this point we have to make some assumptions about motion: since we do not have any dynamical model, we will assume a statistical model. In particular, we will assume that motion is a *first order random walk in* $\mathbb{R}^5$ (see fig. 3.3 left). The problem

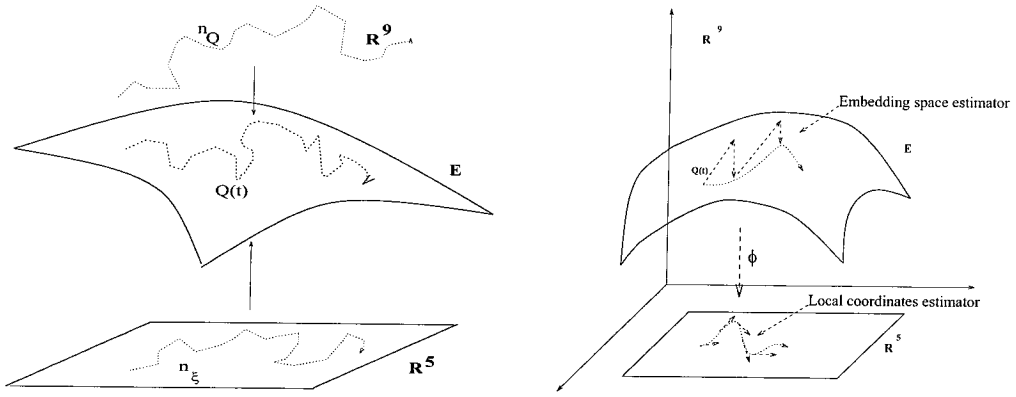

Figure 3.3: *(left) Model of motion as a random walk in* $\mathbb{R}^5$ *lifted to the manifold or as a random walk in* $\mathbb{R}^9$ *projected onto the manifold. (right) Estimation on the Essential space.*

then is to estimate the state of a nonlinear system on a linear space driven by white, zero-mean Gaussian noise (see fig. 3.3 right).

In the second approach we change the model for motion: in particular we assume motion to be a *first order random walk in* $\mathbb{R}^9$ *projected onto the Essential manifold* (fig. 3.3 left). We will see that this leads to a method for estimating motion that consists in solving at each step a *linear estimation* problem in the linear embedding space and then "projecting" the estimate onto the Essential manifold (fig. 3.3 right).

It is very important to understand that these are modeling assumptions about motion which can be validated only a-posteriori. In general we observe that the first method solves a strongly nonlinear problem with techniques which are based upon the linearization of the system about the current reference trajectory, so that the linearization error may be relevant. The second method does not involve any linearization, whereas it imposes the constraint of belonging to the Essential manifold in a weaker way. Note that each method produces, together with the motion estimates, the variance of the estimation error, which can be used to perform recursive triangulation, as in [93].

3.4.1 Estimation in local coordinates

Consider composing the system (3.15) with the map Φ defined in (3.6) restricted to the normalized Essential manifold $\mathbf{E}$:

$$\Phi : \mathbf{E} \rightarrow \mathbf{S}^2 \times \mathbb{R}^3 \rightarrow \mathbb{R}^5$$

$$\mathbf{Q} \mapsto \xi \doteq \begin{bmatrix} T \\ \Omega \end{bmatrix}$$

where T is expressed in spherical coordinates of radius one. Then the system in local coordinates becomes

$$\begin{cases} \xi(t+1) = \xi(t) + n_\xi(t) \ ; \ \xi(t_0) = \xi_0 \\ 0 = \chi_{\mathbf{y}(t),\mathbf{y}'(t)}\mathbf{Q}(\xi(t)) + \tilde{n}(t). \end{cases} \tag{3.17}$$

Motion may be modeled as a first order random walk: $n_\xi(t) \in \mathcal{N}(0, \Sigma_\xi)$ for some Σ_ξ which is referred to as the variance of the model error. While the above assumption is somewhat arbitrary and can be validated only a posteriori, it is often safe to assume that the noise in the measurements $\mathbf{y}(t), \mathbf{y}'(t)$ are white zero-mean Gaussian processes with variance Σ_n. The second order statistics of the induced noise $\tilde{n}$ is a somewhat delicate issue that is discussed in appendix F.4.

The estimation scheme for the model above, which takes into account the correlation of the error $\tilde{n}$, is reported in appendix F.4. A simplified version is obtained by approximating $\tilde{n}$ with a white process (note that $\tilde{n}$ is correlated only within one time step). The resulting scheme is based upon an Implicit Extended Kalman Filter (IEKF), which is derived in appendix F.4. We summarize here the equations of the estimator. Call $C \doteq \left(\frac{\partial \chi \mathbf{Q}}{\partial \xi}\right)$ and $D \doteq \left(\frac{\partial \chi \mathbf{Q}}{\partial \mathbf{x}}\right)$, then we have

Prediction step:

$$\hat{\xi}(t+1|t) = \hat{\xi}(t|t) \ ; \ \hat{\xi}(0|0) = \xi_0 \tag{3.18}$$

$$P(t+1|t) = P(t|t) + \Sigma_\xi \ ; \ P(0|0) = P_0 \tag{3.19}$$

Update step:

$$\hat{\xi}(t+1|t+1) = \hat{\xi}(t+1|t) - L(t+1)\chi(t+1)\mathbf{Q}(\hat{\xi}(t+1|t)) \tag{3.20}$$

$$P(t+1|t+1) = \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + L(t+1)\Sigma_{\tilde{n}}(t+1)L^T(t+1) \tag{3.21}$$

Gain:

$$L(t+1) = P(t+1|t)C^T(t+1)\Lambda^{-1}(t+1) \tag{3.22}$$

$$\Lambda(t+1) = C(t+1)P(t+1|t)C^T(t+1) + \Sigma_{\tilde{n}}(t+1) \tag{3.23}$$

$$\Gamma(t+1) = I - L(t+1)C(t+1) \tag{3.24}$$

Residual variance:

$$\Sigma_{\tilde{n}}(t+1) = D(t+1)\Sigma_n D^T(t+1). \tag{3.25}$$

Note that $P(t|t)$ is the variance of the motion estimation error which is used as variance of measurement error by any "structure from known motion" module [93]. A similar formulation of the IEKF was used by Di Bernardo et al. [26]. Similar expressions were also used before in the literature on specific applications; the first instance to our knowledge was in the recursive computation of the Hough transform [23].

3.4.2 Estimation in the embedding space

Suppose that motion, instead of being a random walk in $\mathbb{R}^5$, is represented in the Essential manifold as the "projection" of a random walk through $\mathbb{R}^9$ (fig. 3.3 left).

We define the operator $\oplus$ that takes two elements in $\mathbb{R}^{3\times3}$, sums them and then projects the result onto the Essential manifold:

$$\begin{aligned} \oplus : \mathbb{R}^{3\times3} \times \mathbb{R}^{3\times3} &\rightarrow E \\ M_1, M_2 &\mapsto \mathbf{Q} = pr_{<E>}(M_1 + M_2), \end{aligned}$$

where the symbol "+" is the usual sum in $\mathbb{R}^{3\times3}$. With the above definitions our model for motion becomes simply

$$\mathbf{Q}(t+1) = \mathbf{Q}(t) \oplus n_{\mathbf{Q}}(t), \tag{3.26}$$

where $n_{\mathbf{Q}}(t) \in \mathcal{N}(0, \Sigma_{n_{\mathbf{Q}}})$ is a white zero-mean Gaussian noise in $\mathbb{R}^9$. If we substitute the above equation into (3.15), we have again a dynamical model on a Euclidean space (in our case $\mathbb{R}^9$) driven by white noise. The Essential estimator is the least variance filter for the above model, and corresponds to a linear Kalman filter update in the embedding space, followed by a projection onto the Essential manifold. In principle, an approximate gain could be precomputed offline for each possible configuration of motion and feature

positions.

Prediction step:

$$\hat{\mathbf{Q}}(t+1|t) = \hat{\mathbf{Q}}(t|t) \; ; \; \hat{\mathbf{Q}}(0|0) = \mathbf{Q}_0 \tag{3.27}$$

$$P(t+1|t) = P(t|t) + \Sigma_{\mathbf{Q}} \; ; \; P(0|0) = P_0 \tag{3.28}$$

Update step:

$$\hat{\mathbf{Q}}(t+1|t+1) = \hat{\mathbf{Q}}(t+1|t) \oplus L(t+1)\chi(t+1)\hat{\mathbf{Q}}(t+1|t) \tag{3.29}$$

$$P(t+1|t+1) = \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + L(t+1)\Sigma_{\tilde{n}}(t+1)L^T(t+1) \tag{3.30}$$

Gain:

$$L(t+1) = -P(t+1|t)\chi^T(t+1)\Lambda^{-1}(t+1) \tag{3.31}$$

$$\Lambda(t+1) = \chi(t+1)P(t+1|t)\chi^T(t+1) + \Sigma_{\tilde{n}}(t+1) \tag{3.32}$$

$$\Gamma(t+1) = I - L(t+1)\chi(t+1). \tag{3.33}$$

3.4.3 Iterated Essential filter

The IEKF update seen in the previous section may be substituted with a Gauss-Newton iteration, as it is customary in recursive ID of linear models:

$$\hat{\xi}(k+1) = \hat{\xi}(k) - L_{NR}(k)h(\hat{\xi}(k))$$

where $L_{NR} = J_h^{-1}(\hat{\xi}(k))$ and J_h is the jacobian of h and $h = \chi\mathbf{Q}$ is the epipolar constraint.

Note that at each fixed time we could perform a Newton-Raphson iteration on the function $h(\mathbf{y}, \xi)$, for which local convergence results can be derived as well as bounds on the convergence rate. This suggests, as an alternative to the IEKF, to fix t and perform a Newton-Raphson iteration along the k coordinate. Once this is done, we propagate the estimate across time with an iteration which now is *linear*, and has all the desirable asymptotic properties.

Iteration at each fixed time

At each time instant a new set of measurements $\mathbf{y}(t)$ becomes available. The coplanarity constraint imposes

$$h\left(\mathbf{y}(t), \xi\right) = 0 \quad \forall t.$$

Define $T_\xi h : \mathbb{R}^m \to \mathbb{R}^n$ to be the derivative of the map h and $J_h(\xi)$ the Jacobian matrix calculated at the point ξ. Suppose that there exists some ξ^* such that $h(\mathbf{y}(t), \xi^*) = 0$ for our particular (fixed) t. Then we may write a first order expansion around the point ξ^*, starting from some point ξ_0 (we neglect time indices for the remainder of this section); the resulting iteration, which is obtained by neglecting the second order term of the expansion, is defined by

$$h\left(\xi_k\right) \doteq J_h(\xi_k)\left(\xi_{k+1} - \xi_k\right).$$

At each iteration we solve for Y the linear problem

$$J_h(\xi_k) Y = h\left(\xi_k\right)$$

and then define $\xi_{k+1} \doteq \xi_k + Y$. In general, also due to noise, we can expect $h\left(\xi_k\right) \notin Im(J_h(\xi_k))$, so that we will be seeking for Y such that $J_h(\xi_k)Y$ is the projection of $h\left(\xi_k\right)$ onto the range space of $J_h(\xi_k)$:

$$\xi_{k+1} \doteq \xi_k - L_{NR}(k) h\left(\xi_k\right).$$

where $L_{NR}(k) \doteq \left(J_h^T(\xi_k) J_h(\xi_k)\right)^{-1} J_h^T(\xi_k)$. The map defined by the right-hand side of the above equation is contractive as long as $J_h(\xi_k)$ has full rank, in which case the scheme is guaranteed to converge to some (possibly local) minimum.

At each time the scheme will converge to some ξ^*, which best explains the noisy measurements $\mathbf{y}^i(t), \mathbf{y}^i(t-1)$; hence we have $\xi^* = \xi + n_\xi$ where n_ξ is an error term, and can be interpreted as a white noise whose variance can be inferred from the variance of n and the linearization of the scheme about zero-noise. The estimate obtained at each fixed time, together with its variance, is fed to a time-integration step, which we

describe next.

Propagation along time

Suppose at each fixed time the iteration along k described above converges to a fixed point $\xi^*(t)$, then we may propagate the information across time with a similar iteration:

$$\hat{\xi}(t+1) = \hat{\xi}(t) + L(t)\left(\xi^*(t) - \hat{\xi}(t)\right)$$

which realizes a linear Kalman filter based upon the model

$$\begin{cases} \xi(t+1) = \xi(t) + n_\xi(t) \\ \xi^*(t) = \xi(t) + n_0(t) \end{cases} \tag{3.34}$$

where n_ξ is the noise driving the random walk model for the parameters, which we assume to be white zero-mean and Gaussian, and n_0 is the error made by the fixed-time iteration. $L(t)$ is the usual linear Kalman gain [58, 55]. The above model has all the desirable properties, as it satisfies the conditions of the asymptotic theorem of Kalman Filtering.

Suppose now that the $k-$iteration has converged to a local minimum, which is compatible with the current observations. At the next step the $t-$iteration will predict an estimate which is in general no longer compatible with the current observations. This should help to disambiguate local minima as the measurements accumulate in time.

3.5 Experimental assessment

In this section we describe an experiment on a real image sequence and one simulation experiment, in order to unravel the different features of each scheme and their behavior when close to singular configurations in the motion space (e.g. pure rotation about the optical axis).

3.5.1 Simulation experiments

We have generated a cloud of 20 feature points at random within a cubic volume of side 1 m, placed 1.5 m ahead of the viewer. The scene was viewed under perspective projection from an image plane of 500×500 pixels with a focal length of 1, corresponding to a visual field of approximately 50^o. Gaussian noise with 1 pixel std was added to the measured projections, according to the performance of the most current feature-tracking schemes [6]. The viewer was then made to navigate around the cloud with constant velocity for 50 time instants (frames), after which the viewer stopped translating and only rotated about its center of projection for 25 frames, inverting the direction after 15 of them. Finally, the viewer resumed its roto-translational motion in order to return to the initial configuration.

This experiment is interesting from many extents: first of all, for part of the sequence the model is in a singular configuration, since the translational velocity is zero. Indeed, as we have discussed in section 3.3.3, the schemes proposed still recover some normalized direction of translation, and the correct rotational velocity. Once the appropriate scaling information has been inserted, full translation is correctly estimated. Secondly, in the first and the last part of the experiment, the motion is designed such that the effects of translation and rotation produce the same variation, up to first order, in the derivative of the observations. This is a well-known ambiguous stimulus in which it is difficult to distinguish locally the effects of rotation from those of translation.

We have systematically varied the conditions of the experiments, by changing the distance in space from the cloud of dots between 1 m and 5 m , the initial conditions between 0% and 1000% off the true value, the level of measurement noise between 0 and 2 pixels and the number of visible points between 1 and 100.

It is interesting to notice that, while previous schemes based upon the Essential matrix needed at least 8 [73] or 5 [51] visible points at each time instant, here we can allow any number of points even below the threshold of 5, since we integrate over time the motion information.

The behavior of the various versions of the Essential filter was consistent, with a graceful degradation of the estimates as the noise level increases, and a need for more precise initial conditions as the noise increases and the number of visible points diminishes. The performance of the filter saturates as the number of visible points increases beyond 20. The performance also degrades as the points move far away from the viewer and as the structure approaches a plane. Under these conditions, in fact, the matrix χ approaches rank 6,

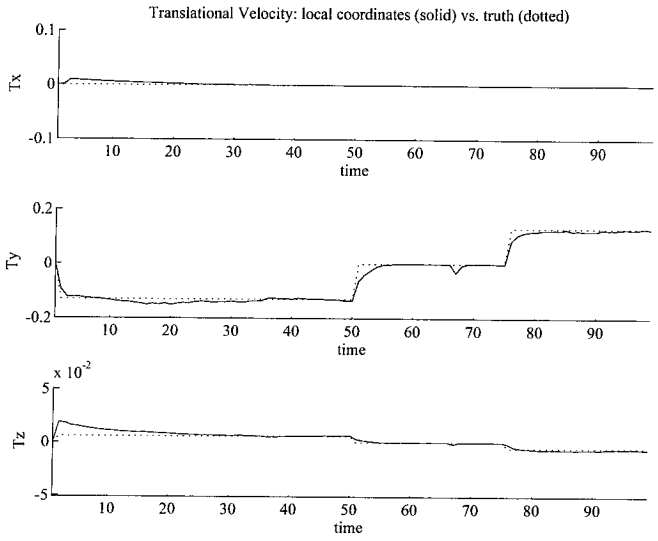

Figure 3.4: *Components of translational velocity as estimated by the local coordinate estimator. The ground truth is shown in dotted lines.*

rather than its normal rank of 8 [33]. A thorough experimental evaluation is reported in chapter 11.

The local coordinate estimator

In figures 3.4-3.5 we show the six components of translational and rotational velocity as estimated by the local coordinates estimator. Ground truth is plotted in dotted lines. Convergence is reached in less than 20 steps from an initial condition within 20% off the true state. Initialization is performed using one step of the traditional Longuet-Higgins' algorithm [73]. Tuning of the filter has been performed, as with the other schemes, within an order of magnitude. It must be pointed out that we have observed a better behavior by increasing the variance of the pseudo-innovation. This is due to the fact that the EKF relies on the hypothesis that the measurement noise is white and the linearization error is negligible, while this is often not the case. An increase in the variance of the measurement noise accounts for the residual of the linearization. The computational cost of one iteration is of about 100 Kflops for 20 points.

The estimator in the embedding space

In fig. 3.8 we show the 9 components of the Essential matrix as estimated by the Essential estimator in the embedding space. Since convergence is about 4 times slower than the local coordinate version, but each step

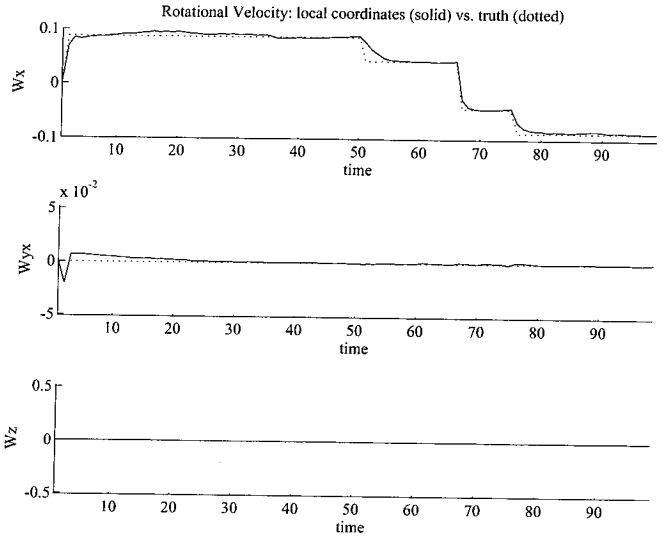

Figure 3.5: *Components of rotational velocity as estimated by the local coordinate estimator.*

requires 4 times less computation, we have sampled the measurements four times faster, ending up with a 400 frames-long sequence.

Note first that, between the frames 200 and 300, the true value of the state is zero. The estimates of the filter drift off to non-zero values, since the Essential matrices are defined as to have unit norm. Such non-zero values are those that allow estimating correctly the rotational velocity and a dummy direction of translation even in the case of pure rotation about the optical axis, as discussed in section 3.3.3. Once transformed the state into local coordinates and inserted the appropriate scale, it is possible to recover the correct rotational and translational components of motion, as shown in figures 3.6-3.7.

The homeomorphism Φ defined in (3.6) may have singularities due to noise when the last eigenspace is exchanged with one of the other two. In fact, in presence of noise, the third singular value of the estimated Essential matrix is non-zero, and occasionally may even become bigger than the other two. Since the SVD sorts the singular values in decreasing order, the eigenvectors – which encode the motion information – may be interchanged.

This causes the spikes observed in the estimates of motion. However, there is no transient to recover, since the errors do not occur in the estimation step, but only in transferring to local coordinates. The switching can be avoided by a higher-level control on the continuity of the singular values. The only significant error in

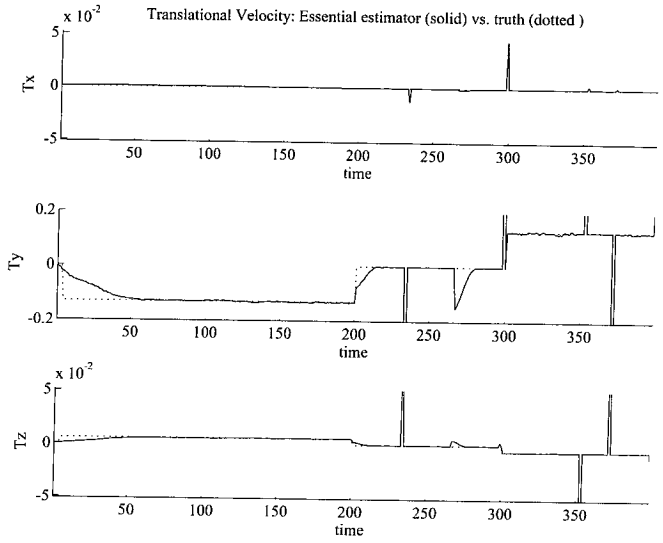

Figure 3.6: *Components of translational velocity as estimated by the Essential estimator. Note the spikes due to the local coordinate transformation. Note also that such spikes do not affect convergence since they do not occur in the estimation process, but while transferring to local coordinates. The switching can be avoided by a higher-level control on the continuity of the singular values of the estimated state. There is a significant error in the local coordinates at around frame 260, when the translation is zero and the direction of rotation is inverted. The smoothness imposed by the dynamics of the parameters is responsible for the transient in the estimates of the rotation, which propagates onto the estimate of translation, causing a visible spike with a significant transient.*

the local coordinates occurs at around frame 260, when the translation is zero and the direction of rotation is inverted. The smoothness imposed by the dynamics of the parameters is responsible for the transient in the estimates of the rotation, which propagates onto the estimates of translation, causing a visible spike with a significant transient. Note that a much less relevant spike was also present in the estimate of the filter in local coordinates (figure 3.4).

The computational cost of our current implementation of the filter in the embedding space amounts to circa 41 Kflops per each step for 20 points. Initialization was performed within 20%, as in the previous case, using one step of the algorithm of Longuet-Higgins [73].

The 2-D iteration

The Essential filter in local coordinates has been implemented using the double iteration described in section 3.4.3. The results are reported in figures 3.9-3.10. This scheme reaches similar accuracy to the local filter

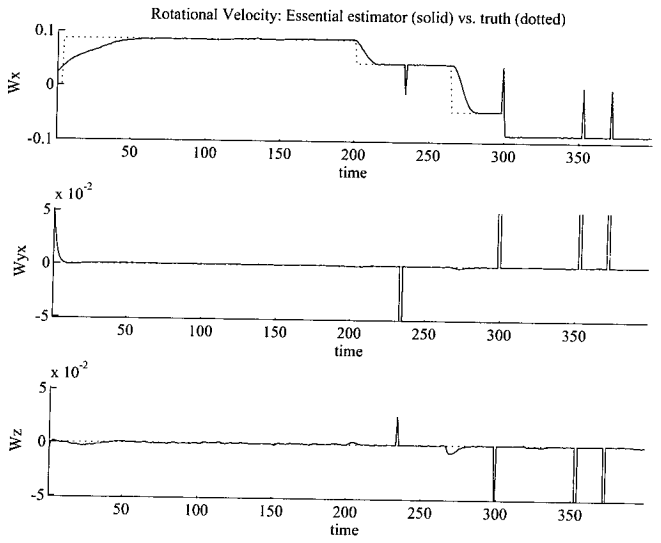

Figure 3.7: *Components of rotational velocity as estimated by the local coordinate estimator. The ground truth is shown in dotted lines. Note the spikes due to the local coordinate transformation. Note also that there is no transient to recover since they do not occur in the estimation process.*

after proper initialization, even though the error analysis used for calculating the variance of the estimates at each fixed time was only approximate. Speed may be adjusted by varying the number of iterations at each fixed time. We have noticed that a number of steps between 3 and 7 is sufficient. The cost of the scheme for 7 iterations and 20 points is 100 Kflops. The simulations reported were performed using a constant variance of the error of the k-iteration.

3.5.2 Experiments on real images

We have tested our schemes on a sequence of 10 images taken at the University of Massachussets at Amherst (see fig 3.11). There are 22 feature points visible; ground truth and feature tracking have been provided. Due to the limited length of the sequence, we have run it on the local coordinates estimator which, however, has a transient of about 10-20 steps to converge from arbitrary initial condition. Hence we have run the local estimator on the 10 images starting from zero initial condition, and we have used the final estimate as initial condition for a new run, whose results we report in figures 3.12-3.14. We did not perform any ad hoc tuning, and the setting was the same used in the simulations described in the previous paragraphs. In fig. 3.12 we report the 6 motion components as estimated by the local coordinate estimator and the corresponding

ground truth (in dotted lines). The estimation error is plotted in figure 3.13. As it can be seen the estimates are within 5% error, and the final estimate is less than 1% off the true motion. Finally, in fig. 3.14 we display the norm of the pseudo-innovation of the filter, which converges to a value of about 10^{-3} in less than $10 + 5$ steps. In this experiment, we have used the true norm of translation as the scale factor.

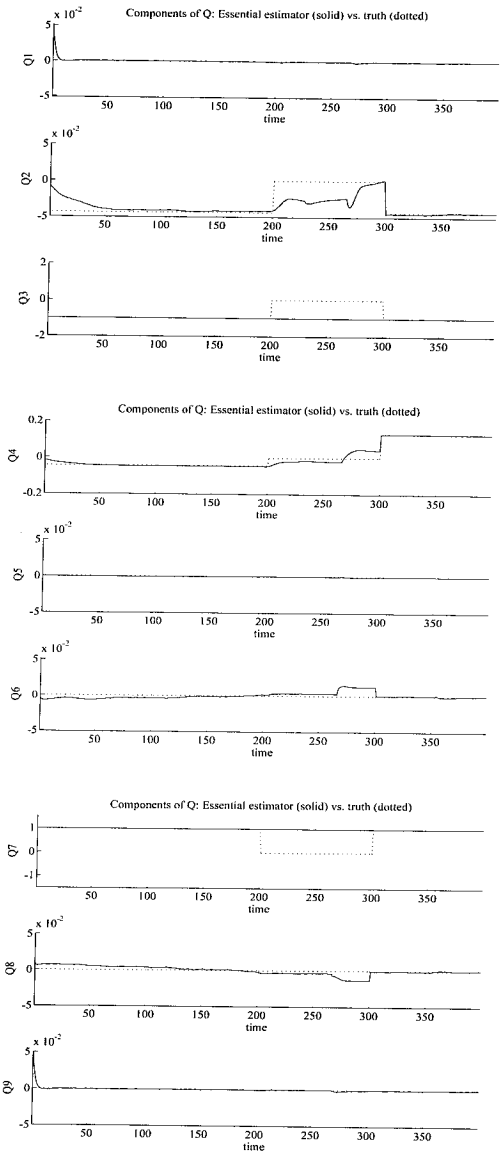

Figure 3.8: *Components of the Essential matrix as estimated by the Essential estimator. Note that there are no spikes. Note that the estimates between time 200 and 300 are non-zero, despite the ground truth (dotted line) is, since the Essential space is normalized to unit-norm. The value of the components of the estimates of* $\mathbf{Q}$ *in the singular region* $T = 0$ *allow us to recover correctly the rotational velocity, once transformed to local coordinates.*

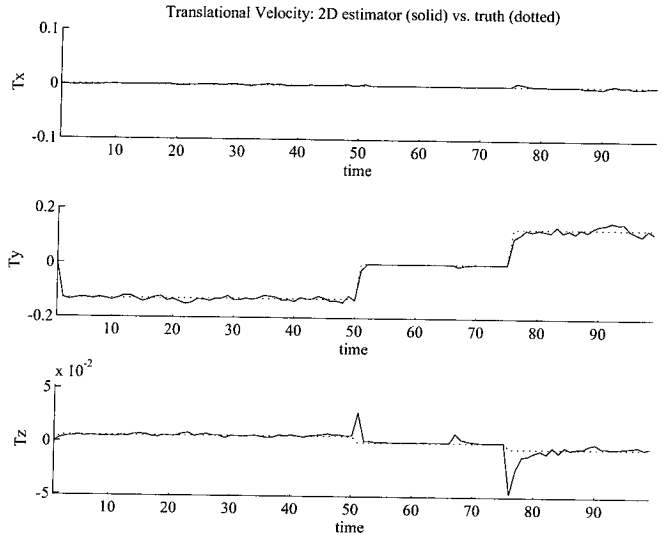

Figure 3.9: *Components of translational velocity as estimated by the double iteration estimator.*

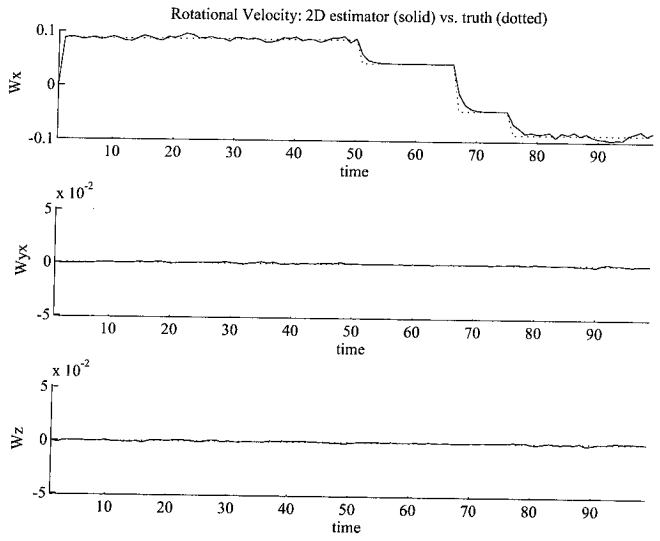

Figure 3.10: *Components of rotational velocity as estimated by the double iteration estimator.*

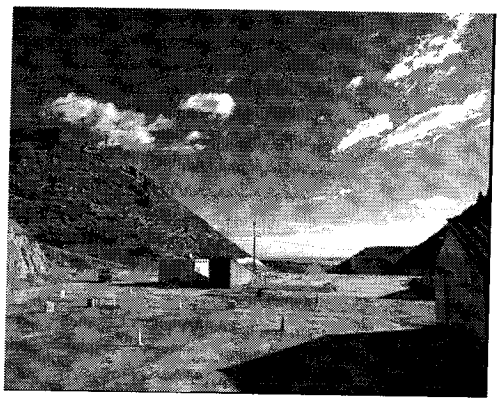

Figure 3.11: *One image of the rocket scene.*

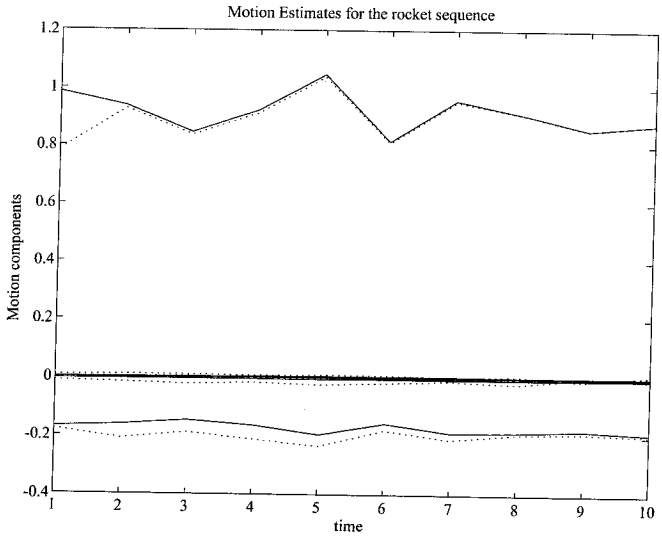

Figure 3.12: *Motion estimates for the rocket sequence: The six components of motion as estimated by the local coordinate estimator are showed in solid lines. The corresponding ground truth is in dotted lines.*

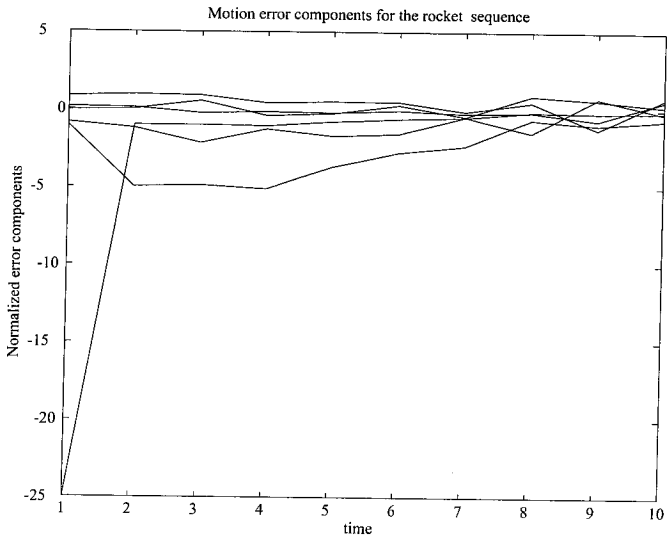

Figure 3.13: *Error in the motion estimates for the rocket sequence. All components are within 5% of the true motion.*

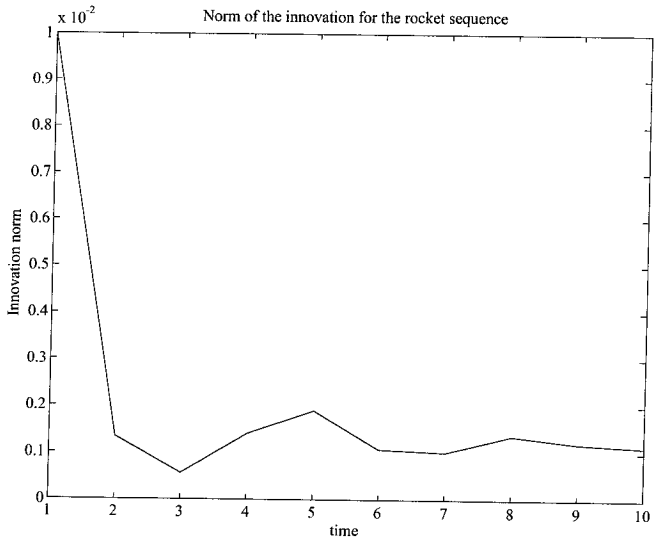

Figure 3.14: *Norm of the pseudo-innovation process of the local estimator for the rocket scene. Convergence is reached in less than 5 steps.*

Chapter 4 Observability of "Structure From Motion"

In this chapter we analyze the possibility of inferring three-dimensional structure and/or motion from a sequence of images. As we have seen in previous chapters, structure and/or motion may be described as state or parameters of nonlinear dynamical models that have perspective projection as a measurement equation. Studying the possibility of inferring structure and/or motion from a sequence of images is equivalent to assessing the observability of the corresponding models.

Background and notation

We are going to use some of the properties of rigid motions, which are described as points on the Lie group $SE(3)$. A rigid motion is composed of a translation vector $T \in \mathbb{R}^3$ and a rotation matrix $R \in SO(3)$. A rotation matrix can be represented locally via the exponential of a three-dimensional vector $\Omega \in \mathbb{R}^3$: $R = e^{\Omega\wedge}$. We could also represent a rigid motion by an instantaneous translational velocity $V \in \mathbb{R}^3$ and a rotational velocity $\omega \in \mathbb{R}^3$ (the infinitesimal generators of the group action). For more details see [81]. For a general introduction of the terminology on observability, see appendix F.

In this chapter we will use the same notation described in chapter 2 and chapter 3. In particular, we call $\mathbf{X} \in \mathbb{R}^3$ the coordinates of a generic three-dimensional vector in space. The coordinates of N points may be collected into a $3 \times N$ matrix $\bar{\mathbf{X}} \in \mathbb{R}^{3\times N}$. The same applies for $\mathbf{y} \in \mathbb{R}\mathrm{P}^2$, which is defined as the perspective projection of the generic point in 3-D: $\mathbf{y} = \pi(\mathbf{X})$. Again, we will use the same symbol for the coordinates of the projection onto a plane and the homogeneous projective coordinates.

Outline of the chapter

We will first study the observability of the models described in chapter 2, and prove that such models are locally observable modulo a one-dimensional subspace of the state-space. This does not mean that the linearization of these models is observable, as we will show. We then study the observability of the Essential model, as described in chapter 2. Reduction does not alter the dimensions of the observable subspace, but it introduces singular configurations in the observable space.

4.1 Observability of structure and motion

In chapter 2 we have seen a number of models involving structure (shape and pose) and motion. Such models are equivalent, in the sense that they are related by coordinate transformations of the state space. Therefore,

we choose one of them to perform the analysis, namely the structure-velocity model

$$\begin{cases} \dot{\mathbf{X}}_0^i = 0 & \mathbf{X}^i(0) = \mathbf{X}_0^i \in \mathbb{R}^3 \\ \dot{v} = 0 & v(0) = v_0 \in \mathbb{R}^6 \qquad \forall\ i\ = 1 \dots N. \\ \mathbf{y}^i(t) = \pi \left(\int_0^t e^{v\wedge\tau} d_{SE(3)}\tau \mathbf{X}_0^i \right) \end{cases} \tag{4.1}$$

We restrict our analysis to constant-velocity motion, and for now we let each point be described by its 3-D coordinates $\mathbf{X}_0^i \in \mathbb{R}^3$ relative to the initial time-instant (we address normalization in section 4.1.1). We also assume that points are in *general position*, which means that they do not line up in zero-measure sets in $\mathbb{R}^3$, and that there is no noise in the measurements. Let

$$\mathbf{y}(t, \mathbf{X}_0, v_0) \tag{4.2}$$

be the (measured) output trajectories starting from initial conditions $\mathbf{X}_0, v_0$. We recall that the generalized velocity v is described in coordinates by a translational velocity V and an instantaneous rotational velocity ω:

$$v\wedge \in se(3) \equiv \begin{bmatrix} V \\ \omega \end{bmatrix} \in \mathbb{R}^6. \tag{4.3}$$

The integral of the generalized velocity gives motion; we write *symbolically*[1]

$$g(t) \doteq \int_0^t e^{v\wedge\tau} d_{SE(3)}\tau \tag{4.4}$$

where $g(t) \in SE(3)$ can be described using a rotation matrix $R(t) \in SO(3)$ and a translation vector $T(t) \in \mathbb{R}^3$, which are given, under constant-velocity, by

$$R(t) = e^{\omega\wedge t} \tag{4.5}$$

$$T(t) = \mathcal{T}(\omega, t)V \tag{4.6}$$

$$\text{where } \mathcal{T}(\omega, t) = (I - e^{\omega\wedge t})\frac{\omega \wedge t}{\|\omega\|^2} + \frac{\omega\omega^T}{\|\omega\|^2}t \qquad \text{if } \|\omega\| \neq 0$$

$$\text{and } \mathcal{T}(0, t) = It.$$

[1] See the discussion in chapter 2 regarding the integral of a generalized velocity.

In this context, the symbol $\wedge$ stands for the cross-product [2]. Therefore, we may write the structure-velocity model in coordinates as

$$\begin{cases} \dot{\mathbf{X}}_0^i = 0 & \mathbf{X}^i(0) = \mathbf{X}_0^i \in \mathbb{R}^3 \\ \dot{V} = 0 & V(0) = V_0 \in \mathbb{R}^3 \\ \dot{\omega} = 0 & \omega(0) = \omega_0 \in \mathbb{R}^3 \\ \mathbf{y}^i(t) = \pi \left(e^{\omega \wedge t} \mathbf{X}_0^i + \mathcal{T}(\omega, t) V \right) \end{cases} \quad \forall\, i \ = 1 \ldots N. \tag{4.7}$$

And now the observability question:

> *Given an initial condition for the above model:* $\bar{\mathbf{X}}_0, V_0, \omega_0$ *and a set of output trajectories* $\{\bar{\mathbf{y}}(t, \bar{\mathbf{X}}_0, V_0, \omega_0)\}_{t \in [0,\tau]}$, *do there exist different choices of initial conditions* $\bar{\mathbf{X}}_a, V_a, \omega_a$ *such that*
>
> $$\bar{\mathbf{y}}(t, \bar{\mathbf{X}}_0, V_0, \omega_0) = \bar{\mathbf{y}}(t, \bar{\mathbf{X}}_a, V_a, \omega_a) \ \forall\, t \in [0, \tau] \ \tau > 0?$$

We call $I(\bar{\mathbf{X}}_0, V_0, \omega_0)$ the set of initial conditions that generate trajectories that are *indistinguishable* from those generated by $\bar{\mathbf{X}}_0, V_0, \omega_0$:

$$I(\bar{\mathbf{X}}_0, V_0, \omega_0) \doteq \{\bar{\mathbf{X}}, V, \omega \mid \bar{\mathbf{y}}(t, \bar{\mathbf{X}}, V, \omega) = \bar{\mathbf{y}}(t, \bar{\mathbf{X}}_0, V_0, \omega_0) \ \forall\, t\}. \tag{4.8}$$

Then the observability question can be written as

$$I(\bar{\mathbf{X}}_0, V_0, \omega_0) \stackrel{?}{=} \{\bar{\mathbf{X}}_0, V_0, \omega_0\}. \tag{4.9}$$

4.1.1 Global observability and the scale ambiguity

The answer to the observability question is clearly negative, since

$$I(\bar{\mathbf{X}}_0, V_0, \omega_0) \supset \{\alpha \bar{\mathbf{X}}_0, \alpha V_0, \omega_0 \mid \forall\, \alpha \in \mathbb{R}\}. \tag{4.10}$$

[2] The cross product between two vectors in $\mathbb{R}^3$, $\mathbf{X}^1 \wedge \mathbf{X}^2$ can be represented as the product of the matrix $(\mathbf{X}^1 \wedge) \doteq \begin{bmatrix} 0 & -X_3^1 & X_2^1 \\ X_3^1 & 0 & -X_1^1 \\ -X_2^1 & X_1^1 & 0 \end{bmatrix}$ with the vector $\mathbf{X}^2$.

This is a well-known scale ambiguity, which essentially says that objects with the same shape "which are twice as big, twice as far and translate twice as fast look identical". To see this it is sufficient to notice that

$$\pi(\bar{\mathbf{X}}) = \pi(\alpha\bar{\mathbf{X}}) \; \forall \; \alpha \in \mathbb{R}, \tag{4.11}$$

and, therefore,

$$\bar{\mathbf{y}}(t, \bar{\mathbf{X}}_0, V_0, \omega) = \pi\left(e^{\omega\wedge t}\bar{\mathbf{X}}_0 + \mathcal{T}(\omega, t)V\right) = \pi\left(e^{\omega\wedge t}\alpha\bar{\mathbf{X}}_0 + \mathcal{T}(\omega, t)\alpha V\right) = \bar{\mathbf{y}}(t, \alpha\bar{\mathbf{X}}_0, \alpha V_0, \omega). \tag{4.12}$$

One way to get rid of such an ambiguity is to *normalize* the states that are affected. One may constrain any one coordinate of $\mathbf{X}^i$ to be a specified value, or constrain the coordinates of the translational velocity, for instance $\|V\| = 1$. Alternatively, we may constrain $\bar{\mathbf{X}}$ to be scaled to norm one. This is done in the models described in the previous chapters by imposing $\bar{\mathbf{X}}_0 \in \mathbf{S}^{3N-1}$, i.e. by choosing $\alpha = \frac{1}{\|\bar{\mathbf{X}}_0\|}$ (see appendix 2.6). Therefore, we know that there will be at least a one-dimensional unobservable subspace of the state of our model, so we may downgrade our observability goal is to see whether this is the only one:

$$I(\bar{\mathbf{X}}_0, V_0, \omega_0) \stackrel{?}{=} \{\alpha\bar{\mathbf{X}}_0, \alpha V_0, \omega_0 \mid \forall \; \alpha \in \mathbb{R}\}. \tag{4.13}$$

4.1.2 Local observability: special cases

Pure rotation

Suppose that motion consists of pure rotation about the optical center, so that $V = 0 \; \forall \; t$. Then it is immediate to see that

$$\mathbf{y}^i(t, \mathbf{X}_0^i, 0, \omega) = \pi\left(e^{\omega\wedge t}\mathbf{X}_0^i\right) = \pi\left(e^{\omega\wedge t}\alpha^i\mathbf{X}_0^i\right) \tag{4.14}$$

and therefore

$$I(\mathbf{X}_0^i, 0, \omega) \supset \{\alpha^i\mathbf{X}_0^i, 0, \omega \mid \forall \; \alpha^i \in \mathbb{R} \; \forall \; i\}. \tag{4.15}$$

Note that this is not an overall scale affecting all points but, rather, one scale parameter per each point. This means that the position in space of each point can be determined only up to a line passing through the center of projection. Its position along this line (depth), however, cannot be recovered. This is also a

well-known fact: pure rotational motion about the optical center does not allow to infer 3-D information about the environment.

We may ask if these are *all* the undistinguishable initial conditions, or if there are other $\omega_a \neq \omega_0$ that generate the same measurements. The answer is no. In fact, if $\mathbf{y}^i(t, \mathbf{X}_0^i, 0, \omega_0) = \mathbf{y}^i(t, \mathbf{X}_0^i, 0, \omega_a)\ \forall\ t$ then, in particular, $\mathbf{y}^i(0, \mathbf{X}_0^i, 0, \omega_0) = \mathbf{y}^i(0, \mathbf{X}_0^i, 0, \omega_a)$, and since $\mathbf{y}^i(0) = \pi(\mathbf{X}_0^i)$, then we can write

$$\mathbf{y}^i(t) = \pi\left(e^{\omega \wedge t}\mathbf{X}_0^i\right) = \pi\left(e^{\omega_0 \wedge t}\mathbf{y}^i(0)\right) = \pi\left(e^{\omega_a \wedge t}\mathbf{y}^i(0)\right) \tag{4.16}$$

and furthermore $\dot{\mathbf{y}}^i(t)_{t=0} = \omega_0 \wedge \mathbf{y}^i(0) = \omega_a \wedge \mathbf{y}^i(0)$, which is true only if

$$\mathbf{y}^i(0) \wedge [\omega_0 - \omega_a] = 0\ \forall\ i. \tag{4.17}$$

That is to say that the difference between the two initial conditions must be parallel to the lines passing through the center of projection and each point on the image-plane. This of course cannot be true as soon as two non-coincident points are observed, which implies that $\omega_a = \omega_0$ and, therefore,

$$I(\mathbf{X}_0^i, 0, \omega) = \{\alpha^i \mathbf{X}_0^i, 0, \omega \mid \forall\ \alpha^i \in \mathbb{R}\ \forall\ i\ = 1 \ldots N \geq 2\}. \tag{4.18}$$

If only one point is observed, then $I(\mathbf{X}_0, 0, \omega) = \{\alpha \mathbf{X}_0, 0, \omega + \lambda \mathbf{X}_0 \mid \forall\ \alpha, \lambda \in \mathbb{R}\}$.

Pure translation

Let us now assume that $\omega = 0\ \forall\ t$, while $V \neq 0$ and characterize the set $I(\mathbf{X}_0^i, V, 0)$. Assume that $\mathbf{y}^i(t, \mathbf{X}_0^i, V_0, 0) = \mathbf{y}^i(t, \mathbf{X}_a^i, V_a, 0)\ \forall\ t$. Then in particular $\mathbf{y}^i(0) = \pi(\mathbf{X}_0^i) = \pi(\mathbf{X}_a^i)$ so that $\mathbf{X}_0^i = \mathbf{y}_0^i Z_0^i$ and $\mathbf{X}_a^i = \mathbf{y}_0^i Z_a^i$. Then we have

$$\mathbf{y}^i(t, \mathbf{X}_0^i, V_0, 0) = \pi\left(\mathbf{y}_0^i Z_0^i + V_0 t\right) = \pi\left(\mathbf{y}_0^i Z_a^i + V_a t\right) \tag{4.19}$$

and so for its derivative at $t = 0$, which can be written as

$$\dot{\mathbf{y}}^i(t)_{t=0} = \mathcal{A}^i \frac{V_0}{Z_0^i} = \mathcal{A}^i \frac{V_a}{Z_a^i}\ \forall\ i\ = 1 \ldots N \tag{4.20}$$

where

$$\mathcal{A}^i \doteq \begin{bmatrix} I & -\mathbf{y}_0^i \end{bmatrix}. \tag{4.21}$$

Therefore, if V_a and Z_a^i generate the same set of measurement as V_0 and Z_0^i, then

$$\mathcal{A}^i \begin{bmatrix} \frac{V_0}{Z_0^i} - \frac{V_a}{Z_a^i} \end{bmatrix} = 0 \; \forall \; i \tag{4.22}$$

which happens either when $\frac{V_0}{Z_0^i} - \frac{V_a}{Z_a^i} = 0$, or when it belongs to the null space of $\mathcal{A}^i \; \forall \; i$. It is immediate to see that the first condition corresponds to $V_a = \alpha V_0$ and $Z_a^i = \alpha Z_0^i$, which is the well-known scale ambiguity. As for the second condition, it corresponds to

$$\frac{V_0}{Z_0^i} - \frac{V_a}{Z_a^i} = \lambda^i \mathbf{y}_0^i \; \forall \; i = 1 \ldots N; \; \lambda^i \in \mathbb{R} \tag{4.23}$$

which can be written, after defining $p^i \doteq \frac{Z_0^i}{Z_a^i}$, as

$$V_0 - p^i V_a = \lambda^i \mathbf{X}_0^i \; \forall \; i = 1 \ldots N; \; \lambda^i, p^i \in \mathbb{R}. \tag{4.24}$$

For the above condition to be satisfied with $\lambda^i \neq 0$ all points $\mathbf{X}_0^i$ must be aligned on plane. In fact, the left hand-side of the above equation describes a number N of points on a line passing through V_0 and parallel to V_a. In order for us to be able to choose scalars λ^i so that $\lambda^i \mathbf{X}_0^i$ line up, the points $\mathbf{X}_0^i$ must belong to a plane through the origin. Since we have assumed *generic-position* conditions, not all points lie on a plane, and therefore the only solution is $\lambda^i = 0$, which again corresponds to the usual scale ambiguity. Therefore we have

$$I(\bar{\mathbf{X}}_0, V, 0) = \{\alpha \bar{\mathbf{X}}_0, \alpha V \; \forall \; \alpha \; \in \mathbb{R}\}. \tag{4.25}$$

Planar structure

From the previous discussion, which will be made general in section 4.1.3, we see that structure and motion are observable – modulo the scale ambiguity – whenever the points do not lie on a plane passing through the origin (optical center). Note that, in order to maintain all given points on a plane through the origin, we need to constrain motion so that the optical center remains on that plane, which means that we can only

rotate about a vector orthogonal to the plane, and translate along a vector parallel to the plane. Therefore, *if we know that motion and structure are on some plane*, we can take it as a slice of a cylindrical world, which projects onto an image-line, and re-scale the whole problem to a "2-D from 1-D" task. We can easily derive models which are equivalent to the ones proposed in section 2.2.1 by just substituting

$$\begin{aligned} \mathbb{R}^3 &\longrightarrow \mathbb{R}^2 \\ \mathbb{RP}^2 &\longrightarrow \mathbb{RP}^1 \\ \mathbf{S}^2 &\longrightarrow \mathbf{S}^1 \\ SE(3), se(3) &\longrightarrow SE(2), se(2) \\ \Sigma^N = \mathbf{S}^{3(N-1)-1} \backslash SO(3) &\longrightarrow \mathbf{S}^{2(N-1)-1} \backslash SO(2). \end{aligned}$$

Therefore, if we know that motion and structure lie on a plane, we can eliminate the un-necessary states and re-formulate the problem in a smaller-dimensional space. Note that this situation actually occurs, for instance, in autonomous navigation in buildings' interiors, and it can be readily detected using simple rank tests [106].

Another interesting case is when points are contained on a plane *not passing through the optical center*, i.e.

$$\mathbf{X}^i \in \mathbb{R}^3 \mid \exists\, \mathbf{n} \in \mathbb{R}^3 \mid \mathbf{n}^T \mathbf{X}^i = 1. \tag{4.26}$$

In such a case the transformation undergone by the projection of feature points between any two different views can be represented as an homography, which consists of a scaled linear transformation of the projective coordinates:

$$\mathbf{y}^i(t) \sim A(t)\mathbf{y}_0^i \in \mathbb{RP}^2 \tag{4.27}$$

where $\sim$ indicates equality up to a scaling factor and $A(t)$ is a generic 3×3 matrix. In order to see that it is sufficient to write the measurement equation as

$$\mathbf{y}^i(t) = \pi\left(R(t)\mathbf{X}_0^i + T(t) \cdot 1\right) = \pi\left(R(t)\mathbf{X}_0^i + T(t)\mathbf{n}^T\mathbf{X}_0^i\right) = \pi\left(A(t)\mathbf{X}_0^i\right) \tag{4.28}$$

where we have defined $A(t) \doteq R(t) + T(t)\mathbf{n}^T$. The last equation is equivalent to (4.27), and is *linear* in $A(t)$, which has 8 independent parameters that contain all the information about rotation, scaled translation and normal vector.

Such a case is singular in epipolar geometry [76], while we are going to see soon that it is observable from the models described in section 2.2.1.

4.1.3 The general case

We have seen that shape can be observed at most up to a scale when translation is not identically zero. When there is no rotational velocity, then such a scale ambiguity is the only unobservable subspace under general-position conditions. We now turn to prove that rotation is irrelevant as far as observability of structure is concerned, and therefore we can carry out a reasoning similar to the zero-rotation case to prove that structure is observable modulo a scale subspace under general-position conditions. Before proceeding, we make a few remarks.

Remark 4.1.1 *The observability of structure (modulo a scale) does not imply that there do not exist configurations of points that do not allow reconstruction of motion or structure. It just says that such configurations are a zero-measure set (after scale compensation), and therefore turns their study into a singularity analysis.*

Remark 4.1.2 *The tools used for studying observability are intrinsically* local. *Which means that, given two sets of initial conditions, their corresponding output trajectories can be distinguished by staying arbitrarily close to the initial conditions. This does not imply* in general *that the same output trajectories can be distinguished for arbitrary state evolutions, for one could conceive some periodic motions and a synchronous output such that the initial conditions become indistinguishable. However, again, these are pathological conditions, while under* generic conditions *we can assume that we can distinguish output trajectories corresponding to different initial conditions for arbitrary state evolutions.*

Remark 4.1.3 *Note that local observability does not imply that the linearization of the model is observable, as we will show in section 4.1.5.*

We now proceed with proving that

$$I(\bar{\mathbf{X}}_0, V_0, \omega_0) \subset \{\alpha\bar{\mathbf{X}}_0, \alpha V_0, \omega_0 \ \forall \ \ \alpha \in \mathbb{R}\}. \tag{4.29}$$

As we have done in the zero-rotation case, we exploit the measurements at time $t = 0$ to substitute $\mathbf{X}_0^i = \mathbf{y}_0^i Z_0^i$ and $\mathbf{X}_a^i = \mathbf{y}_0^i Z_a^i$. Then we have

$$\dot{\mathbf{y}}^i(t)_{t=0} = \mathcal{A}^i \frac{V_0}{Z_0^i} + \mathcal{B}^i \omega_0 = \mathcal{A}^i \frac{V_a}{Z_a^i} + \mathcal{B}^i \omega_a \quad \forall\, i \ = 1 \ldots N \tag{4.30}$$

where

$$\mathcal{B}^i \doteq \begin{bmatrix} -\mathbf{y}_1^i \mathbf{y}_2^i & 1 + (\mathbf{y}_1^i)^2 & -\mathbf{y}_1^i \\ -1 - (\mathbf{y}_2^i)^2 & \mathbf{y}_1^i \mathbf{y}_2^i & \mathbf{y}_2^i \end{bmatrix}_{t=0}. \tag{4.31}$$

We can write the above equation as

$$\mathcal{A}^i \left(\frac{V_0}{Z_0^i} - \frac{V_a}{Z_a^i}\right) = \mathcal{B}^i \tilde{\omega} \ \forall\, i \ = 1 \ldots N \tag{4.32}$$

where $\tilde{\omega} = \omega_a - \omega_0$. For this to be true it is necessary that the vector $\mathcal{B}^i \tilde{\omega}$ be in the range space of $\mathcal{A}^i$ for all i, i.e. there must be scalars $\lambda_1^i, \lambda_2^i, \lambda_3^i$ such that

$$\mathcal{A}^i \begin{bmatrix} \lambda_1^i \\ \lambda_2^i \\ \lambda_3^i \end{bmatrix} = \mathcal{B}^i \tilde{\omega} \tag{4.33}$$

or equivalently

$$\begin{bmatrix} \lambda_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\mathbf{y}_1^i \mathbf{y}_2^i \\ -1 - {\mathbf{y}_2^i}^2 \end{bmatrix} \tilde{\omega}_1$$

$$\begin{bmatrix} 0 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 1 + {\mathbf{y}_1^i}^2 \\ \mathbf{y}_1^i \mathbf{y}_2^i \end{bmatrix} \tilde{\omega}_1$$

$$\begin{bmatrix} -\mathbf{y}_1^i \lambda_3 \\ -\mathbf{y}_2^i \lambda_3 \end{bmatrix} = \begin{bmatrix} -\mathbf{y}_2^i \\ \mathbf{y}_1^i \end{bmatrix} \tilde{\omega}_1.$$

The first set of equations imply that $\tilde{\omega}_1 = 0$ or ${\mathbf{y}_2^i}^2 = -1$, similarly the second set implies $\tilde{\omega}_2 = 0$ or ${\mathbf{y}_1^i}^2 = -1$, and the third equation implies either $\omega_3 = 0$ or $\mathbf{y}_1^2 = -\mathbf{y}_2^2$. Therefore we conclude that

$$\tilde{\omega} = 0 \tag{4.34}$$

which means that there are no indistinguishable rotations. This also implies that, if there are undistinguishable combinations of translation and depth, these must satisfy

$$\mathcal{A}^i(\frac{V_0}{Z_0^i} - \frac{V_a}{Z_a^i}) = 0 \; \forall \; i \; = 1 \ldots N \tag{4.35}$$

which brings us back to the analysis of the zero-rotation case.

From the discussion in the previous section, it follows also that $V_a = \alpha V_0$ and $Z_a^i = Z_0^i$, which allows us to conclude that

$$I(\bar{\mathbf{X}}_0, V_0, \omega_0) = \{\alpha\bar{\mathbf{X}}_0, \alpha V_0, \omega_0 \mid \forall \; \alpha \in \mathbb{R}\}. \tag{4.36}$$

Therefore motion and structure are locally observable modulo a one-dimensional linear subspace. Analogous results can be obtained for discrete-time models, under general-position conditions involving structure, motion as well as time sampling.

4.1.4 Local-weak observability

In the previous section we have explored the local observability of structure and motion by assuming general-position conditions. Such conditions are not satisfied when points line up in some particular plane in 3-D or when just one single point is visible.

Since we can measure the image over an extended period of time, we could try to take increasing levels of derivatives, and explore to what extent motion and structure are observable with one point only.

The local observability space $\mathcal{O}$ is defined as the set of the output functions and all their possible Lie derivatives along vector-fields in the accessibility algebra (see [53, 48, 63, 64, 66, 82, 104] for an introduction on nonlinear observability, and appendix F for notation). Under the constant velocity assumption, the state vector-field in (2.35) is autonomous and, therefore, the observability space is spanned by

$\{\pi, L_{\bar{f}}\pi, \ldots, L^k_{\bar{f}}\pi \ldots\}$, where $\bar{f}$ represents the state vector-field and $L_{\bar{f}}\pi$ is the Lie derivative of the output function π along the state vector-field . The observability codistribution is $d\mathcal{O} \doteq \{dh \mid h \in \mathcal{O}\}$. The state manifold is $\mathbb{R}^9$, intended as a local coordinatization of $se(3) \times \mathbb{R}^3$. Computing the observability codistribution is trivial but messy, and we have employed symbolic manipulation programs that also compute normal rank. Such a rank reaches its maximum of 8 after two levels of Lie differentiation (and therefore three derivatives overall). One could therefore conjecture that the one-dimensional unobservable subspace coincides with the scale ambiguity discussed in previous sections.

Indeed it is the case, for it can be verified that the null space of the observability codistribution is, in case of nonzero translation

$$\text{Null}([d\pi, dL_{\bar{f}}\pi, dL^2_{\bar{f}}\pi, dL^3_{\bar{f}}\pi]) = \text{Span}\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 & V_1 & V_2 & V_3 & 0 & 0 & 0 \end{bmatrix}.$$

In the case of pure rotation, a basis of the null space of the observability codistribution is

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and all the points with the same projective coordinates are indistinguishable. In the case of nonzero forward translation, the set of states which are indistinguishable from $[\mathbf{X}_0\ V_0\ \omega_0]^T$ is therefore

$$I(\mathbf{X}_0, V_0, \omega_0) = \{ \begin{bmatrix} \mathbf{X}_0 s \\ V_0 s \\ \omega_0 \end{bmatrix} \mid s \in \mathbb{R}\},$$

which corresponds to what we have found for the general case.

Therefore, even in the case of one single visible point, we find that structure and motion are locally observable modulo the one-dimensional subspace, under the constant velocity assumption. Indeed, adding more points does not change the structure of the observability space, since feature points are not coupled with each other in the state model.

From our point of view, the line of work in epipolar geometry that tries to derive explicitly all constraints involving measurements and unknown parameters is equivalent to an algebraic analysis of the observability

codistribution. The fact that the observability codistribution reaches the maximum rank after three levels of differentiation (and therefore four derivatives overall, since the observability codistribution is the collection of differentials of the Lie derivatives of the output) is equivalent to the statement that there are no independent constraints beyond the quadri-linear ones (in the measurement), derived by Faugeras [31].

One may then raise the question of what is the best way to exploit all independent constraints: whether to use a local observer, which is practically equivalent to a sort of dynamic inverter of the observability codistribution, or to write explicitly all constraints and then solve polynomial equations to estimate the unknown parameters. The answer depends upon the particular application one is targeting, as we have discussed in 2.4.

4.1.5 Linear observability

Consider the model for structure and motion estimation, for instance in its differential form (2.35). If we linearize that model around a reference initial condition, for instance with all points on the image plane, then we get the linear system

$$\begin{cases} \dot{\xi} = A\xi \\ y = C\xi \end{cases} \tag{4.37}$$

where $\xi = \{\bar{\mathbf{X}}, V, \omega\}$, $A \doteq \frac{\partial \bar{f}(\xi)}{\partial \xi}$ $C \doteq \frac{\partial \pi(\xi)}{\partial \xi}$; for $N = 1$ we have

$$\begin{aligned} A &= \begin{bmatrix} (\omega_0\wedge) & I_3 & -(\bar{\mathbf{y}}_0\wedge) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ C &= \begin{bmatrix} \frac{1}{Z}\mathcal{A} & 0 & 0 \end{bmatrix} \\ CA^i &= \frac{1}{Z}\mathcal{A}\begin{bmatrix} (\omega_0\wedge)^i & (\omega_0\wedge)^{i-1} & -(\omega_0\wedge)^{i-1}(\bar{\mathbf{y}}_0\wedge) \end{bmatrix}. \end{aligned}$$

The observability matrix for the linearized system is hence

$$O = \begin{bmatrix} \frac{1}{Z}\mathcal{A} & 0 & 0 \\ \frac{1}{Z}\mathcal{A}(\omega_0\wedge) & \frac{1}{Z}\mathcal{A} & -\frac{1}{Z}\mathcal{A}(\bar{\mathbf{y}}_0\wedge) \\ \vdots & \vdots & \vdots \\ \frac{1}{Z}\mathcal{A}(\omega_0\wedge)^8 & \frac{1}{Z}\mathcal{A}(\omega_0\wedge)^7 & -\frac{1}{Z}\mathcal{A}(\omega_0\wedge)^7(\bar{\mathbf{y}}_0\wedge) \end{bmatrix}.$$

It is easy to see that O has rank 5, in face of a state space of dimension 9. The linearized system is, therefore, not observable, and we say that the original model is not linearly observable.

4.2 Observability of the Essential model

In the framework of the Essential filter, motion estimation is viewed as the problem of identifying the following nonlinear implicit model with parameters on a the "Essential manifold":

$$\begin{cases} \mathbf{x}^i(t+1)^T\mathbf{Q}\mathbf{x}^i(t) = 0 & \mathbf{Q} = T\wedge R \mid (T\wedge) \in so(3), R \in SO(3) \\ \mathbf{y}^i(t) = \mathbf{x}^i(t) + n^i(t) & \forall\, i = 1\ldots N. \end{cases} \tag{4.38}$$

Since the Essential constraint is linear in $\mathbf{Q}$, it is possible to write it using the notation

$$\chi(\mathbf{x}, \mathbf{x}')\mathbf{Q} = 0$$

where χ is a $n \times 9$ matrix and $\mathbf{Q}$ is interpreted as a nine-dimensional vector obtained by stacking the columns of $\mathbf{Q}$ on top of each other. We also write $\mathbf{x}'$ for $\mathbf{x}(t+1)$. We will use both notations $\chi(\mathbf{x}, \mathbf{x}')$ and $\chi(t)$ depending on whether we want to emphasize the points that contribute to the epipolar constraint or the time in which it is computed.

We will assume *constant velocity*, so that the unknown parameters in the Essential model (4.38) are described as a point on the Essential manifold E. We may therefore transform the identification task onto a state estimation task where the parameters and their (trivial) dynamics play the role of the state [3]. The

[3] In a stochastic framework the parameters of the Essential model describe a random walk in the Essential manifold. In chapter 3 Essential parameters are modeled as either a random walk in the local coordinates *lifted* to the Essential manifold, or as a random walk in the embedding space *projected* onto the manifold.

resulting model has the form

$$\begin{cases} \dot{\mathbf{Q}}(t) = 0 & \mathbf{Q} \in E \\ \chi(\bar{\mathbf{y}}, \bar{\mathbf{y}}')\mathbf{Q} = \nu_\chi \end{cases} \tag{4.39}$$

where the noise term ν_χ is induced by having substituted $\mathbf{y}^i$ for $\mathbf{x}^i$ in the Essential constraint. Note that the above model is *independent of structure* (shape and pose), and it only describes inter-frame motion (or "discrete velocity"). It is possible to integrate the above model from the initial time instant, so that $\mathbf{Q}$ describes motion relative to the initial time, just by substituting $\bar{\mathbf{y}}'$ with $\bar{\mathbf{y}}(0)$.

Remark 4.2.1 *The price we must pay in order to eliminate structure is that the noise term ν_χ is* no longer white, *for it is correlated within one time-step. In order to whiten it it is necessary to add N states to the model – which annihilates the benefits of the above model – as it is shown in [90].*

However, a simple implicit Extended Kalman Filter (IEKF) implemented by approximating ν_χ with a white noise has proven effective in real-world situations. In chapter 3, two recursive schemes are proposed for solving the estimation problem: one is based upon an IEKF in the local coordinates of the Essential manifold, the other is based upon a linear update on the linear embedding space $\mathbb{R}^9$, followed by a projection onto the Essential manifold.

We now turn our attention to the observability of the model (4.39). Suppose that, at time $t + \tau_i$, the matrix $\chi(t + \tau_i)$ has a null space of dimension k_i. *When the viewer moves with constant velocity,* we may write

$$\begin{aligned} \chi(t)\mathbf{Q}(t) &= 0 \\ \chi(t+\tau_1)\mathbf{Q}(t+\tau_1) = \chi(t+\tau_1)\mathbf{Q}(t) &= 0 \\ \vdots &= \vdots \\ \chi(t+\tau_p)\mathbf{Q}(t+\tau_p) = \chi(t+\tau_p)\mathbf{Q}(t) &= 0 \end{aligned}$$

and state rank conditions for observability on the extended matrix

$$\bar{\chi}_p \doteq \begin{bmatrix} \chi(t) \\ \chi(t+\tau_1) \\ \vdots \\ \chi(t+\tau_p) \end{bmatrix}.$$

It is easy to prove [4] that the above matrix $\bar{\chi}$ reaches rank 8 if and only if there does not exist a plane that contains all the visible points or a (proper) quadric surface[5] in $\mathbb{R}^3$ which contains all the visible points and the path of the center of projection. It is immediate to see that, if all the visible points are contained in a plane, then χ has rank 6. Suppose therefore that the visible points do not lie on a plane. We treat visible points and centers of projection at subsequent time instants in the same fashion, since they play equivalent roles in the Essential constraint. Therefore, we will choose two of them as reference centers of projection, and treat the remaining as visible points. We now show that rotation plays no role in the study of observability of the Essential model.

Let $T \neq 0$. Consider the epipolar constraint written relative to a reference frame which is centered mid-way between the two optical centers at each time step. Call $\tilde{T} \doteq \frac{T}{2}$, $\tilde{R} \doteq e^{\frac{\Omega}{2}\wedge}$ and $\tilde{\mathbf{X}} = \tilde{R}\mathbf{X} + \tilde{T}$. The Essential constraint reads

$$\mathbf{x'}^{i^T}\mathbf{Q}\mathbf{x}^i = [\tilde{R}(\tilde{\mathbf{x}}^i - \tilde{T})]^T\mathbf{Q}\tilde{R}^T(\tilde{\mathbf{x}}^i + \tilde{T}) = (\tilde{\mathbf{x}}^i - \tilde{T})^T\tilde{R}^T\mathbf{Q}\tilde{R}^T(\tilde{\mathbf{x}}^i + \tilde{T}) = 0 \quad 1 \leq i \leq N. \tag{4.40}$$

Since $\tilde{R}$ is invertible, we may redefine $\mathbf{Q}$ to be $\tilde{R}^T\mathbf{Q}\tilde{R}^T$ without loss of generality. Therefore, we will assume $R = I$. Equation (4.40) becomes

$$(\tilde{\mathbf{x}}^i - \tilde{T})\mathbf{Q}(\tilde{\mathbf{x}}^i + \tilde{T}) = \tilde{\mathbf{x}'}^{i^T}\mathbf{Q}\tilde{\mathbf{x}}^i - \tilde{T}^T\mathbf{Q}\tilde{\mathbf{x}}^i + \tilde{\mathbf{x}'}^{i^T}\mathbf{Q}\tilde{T} - \tilde{T}^T\mathbf{Q}\tilde{T} = 0 \quad 1 \leq i \leq N. \tag{4.41}$$

Call $< \mathbf{Q} > \doteq \{\mathbf{Q} \in \mathbb{R}^{3\times 3} \mid (\tilde{\mathbf{x}}^i - \tilde{T})^T\mathbf{Q}(\tilde{\mathbf{x}}^i + \tilde{T}) = 0, \quad 1 \leq i \leq N\}$, which is a vector subspace of $\mathbb{R}^{3\times 3}$. We have to prove that its dimension is one. Indeed, $\dim(< \mathbf{Q} >)$ is always bigger or equal than one, since

[4] The following result is a more general version of the result stated by Longuet-Higgins for the case of stereo (two views) [74]. A. Mennucci (personal note) provided a version of the proof which has been extended here.

[5] A quadric surface is a set $\{x \in \mathbb{R}^{3\times 3} \mid x^TAx + b^Tx + c = 0\}$ where A is a 3×3 matrix, b is a $3-$vector and c is a scalar. It is proper if it is a proper subset of $\mathbb{R}^3$.

it contains the matrix $\tilde{T}\wedge$, as can be seen by direct substitution in eq. (4.41).

Now, suppose that the equation (4.40) holds for a matrix M, and decompose it in the symmetric and antisymmetric part $A = \frac{M-M^T}{2}, S = \frac{M+M^T}{2}$, then

$$\tilde{\mathbf{x}^i}^T S\tilde{\mathbf{x}}^i - 2\tilde{T}^T A\tilde{\mathbf{x}}^i - \tilde{T}^T S\tilde{T} = 0 \quad 1 \leq i \leq N.$$

Then consider the set $< V > \doteq \{x \in \mathbb{R}^3 \mid x^T Sx - 2\tilde{T}^T Ax - \tilde{T}^T S\tilde{T} = 0\}$. This set always contains the two points $\tilde{T}$ and $-\tilde{T}$, the centers of projection, as it can be verified.
Suppose there is no (proper) quadric surface containing the points $\tilde{\mathbf{x}}^i$; then it must be that $V = \mathbb{R}^3$, that means that $S = 0$ and $\tilde{T}^T A = 0$; this means that M is necessarily a multiple of $\tilde{T}\wedge = \mathbf{Q}$, so we get that $\dim(< \mathbf{Q} >) = 1$.

Vice versa, suppose that the symmetric part S of M is nonzero or that $\tilde{T}^T A \neq 0$; then the set $< V >$ is a quadric surface that contains the points $\tilde{\mathbf{x}}^i$ (by definition), and the points $\tilde{T}$ and $-\tilde{T}$, which are the two centers of projection (if the symmetric part $S = 0$, then the set $\{x \in \mathbb{R}^3 \mid T^T Ax = 0\}$ is a plane, which is in any case a quadric surface).

Remark 4.2.2 *Note that the quadric surface is a thin set in the 3-D Euclidean space, and in general the measurement noise in the projected coordinates is sufficient to set the model in general-position. Note also that $T \neq 0$ plays a critical role in achieving global observability, while Ω (or R) has no influence.*

Remark 4.2.3 *Notice that the elimination of structure from the Essential model has the additional cost of introducing singular configurations that were not present in the general models described in section 2.2.1.*

Chapter 5 Observer reduction in the continuous case: motion estimation from subspace constraints

In previous chapters we have seen how structure and motion can be described as the state of a nonlinear dynamical model that has a perspective projection in the measurement equation. Structure and motion may be estimated simultaneously by a state observer. We have also seen how it is possible to reduce the order of the observer by eliminating structure from the model (chapter 3). In this chapter we show how it is possible to further decouple the translational component of motion from the rotational one, and eliminate the latter so as to be left with a model with only two unknown parameters describing the direction of heading. This can be done easily in a differential (continuous-time) framework, since the motion parameters appear linearly in the model.

Background and notation

We are going to adopt a continuous-time representation of rigid motion, through a translational velocity V and a rotational velocity ω. We call $\mathbf{X} = [X\ Y\ Z]^T \in \mathbb{R}^3$ the coordinates of a generic point in space, $\mathbf{x} = [x\ y]^T = \pi(\mathbf{X})$ (or $\mathbf{x} = [x\ y\ 1]^T$) its perspective projection, as in chapter 2, and $\mathbf{y}$ the noisy version of $\mathbf{x}$: $\mathbf{y} = \mathbf{x} + n$. The velocity of the generic point under the action of a rigid motion is given by $\dot{\mathbf{X}} = \omega \wedge \mathbf{X} + V$, where $\wedge$ denotes the cross product. Due to the presence of the scale-factor ambiguity, we represent the translational velocity V using spherical coordinates θ, ϕ, as we describe in the appendix at the end of this chapter. We use the notation Σ_ξ to denote the variance/covariance matrix of the random vector ξ.

Outline of the chapter

We first describe a technique for decoupling the depth and rotational velocity from the models described in chapter 2. It is based upon a linear projection resulting in the so-called "Subspace constraint", introduced in [45]. Such a constraint was used by Heeger and Jepson to formulate an optimization task in order to estimate the direction of heading from optical flow. We take a different approach, and view the constraint of Heeger and Jepson as a dynamical model of a very peculiar class, that of so-called "Pfaffian Systems" (a particular type of Exterior Differential Systems [16]). Such a model, called the "Subspace model", has as unknown parameter the direction of heading, represented as a point on a sphere. Estimating motion then amounts to identifying the Subspace model.

5.1 Motion reconstruction via least-squares inversion constrained on subspaces

Consider any of the models resulting from the constraint of rigid motion and perspective projection, described in chapter 2. The first derivative of the output of such models, which is referred to in the literature as the "motion field", represents the velocity of the projection of the coordinates of each feature-point in the image-plane:

$$\dot{\mathbf{x}}^i(t) = \left[\frac{1}{Z^i}\mathcal{A}^i \mid \mathcal{B}^i\right] \begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix} \tag{5.1}$$

where

$$\begin{aligned} \mathcal{A}^i &\doteq \begin{bmatrix} 1 & 0 & -x^i \\ 0 & 1 & -y^i \end{bmatrix} \\ \mathcal{B}^i &\doteq \begin{bmatrix} -x^i y^i & 1+{x^i}^2 & -y^i \\ -1-{y^i}^2 & x^i y^i & x^i \end{bmatrix}. \end{aligned} \tag{5.2}$$

The motion field is not directly measurable. Instead, what we measure are brightness values on the imaging sensor. For practical purposes, the motion field is approximated by the "optical flow", which consists in the velocity of brightness patches on the image-plane. Such an approximation is by and large satisfied in the presence of highly textured Lambertian surfaces and constant illumination. However, outliers are quite common in realistic image sequences, due to the presence of occlusions, specularities, shadows etc. . Any motion estimation algorithm willing to operate in real-time on realistic sequences must be able to deal with such situations in an automatic fashion.

In the next sections we will assume that we can measure directly the motion field, neglecting outliers. Only later, in section 5.3.3, will we show how it is possible to spot-out outliers due, for instance, to T-junctions, specularities, matching errors from the feature-tracking algorithm, and reject them before they can affect the estimates of 3-D motion.

5.1.1 Recovery of the direction of translation from two views

By observing a sufficient number of points $\mathbf{x}^i \ \forall \ i = 1 \dots N$, one may use eq. (5.1) for writing an overdetermined system which can be solved for the inverse depth and the rotational velocity in a least-squares fashion. To this end, rearrange equation (5.1) as

$$\dot{\mathbf{x}}^i(t) = \left[\mathcal{A}^i V(\theta, \phi) \mid \mathcal{B}^i\right] \begin{bmatrix} \frac{1}{Z(t)^i} \\ \omega(t) \end{bmatrix}.$$

Since the translational velocity V multiplies the inverse depth of each point, both can be recovered only up to an arbitrary scale factor. Due to this scale ambiguity, we may only reconstruct the direction of translation; hence V may be restricted to be of unit norm, and represented in local (spherical) coordinates[1] as $V(\theta, \phi) \in \mathbf{S}^2$. For instance, θ may denote the azimuth angle in the viewer's reference, and ϕ the elevation angle. If some scale information becomes available, as for example the size of a visible object, it is possible to rescale the depth and the translational velocity, as we will discuss in the experimental section. When N points are visible, the equations above may be rearranged into a vector equality:

$$\dot{\mathbf{x}} = \tilde{\mathcal{C}}(\mathbf{x}, \theta, \phi)[\frac{1}{Z_1}, \dots, \frac{1}{Z_N}, \omega]^T, \tag{5.3}$$

where

$$\tilde{\mathcal{C}}(\mathbf{x}, \theta, \phi) \doteq \begin{bmatrix} \mathcal{A}_1 V & & & \mathcal{B}_1 \\ & \ddots & & \vdots \\ & & \mathcal{A}_N V & \mathcal{B}_N \end{bmatrix}$$

and $\mathbf{x}$ is a $2N$ column vector obtained by stacking the $\mathbf{x}^i \ \forall \ i = 1 \dots N$ on top of each other. At this point one could solve the above equation (5.3) in a least-squares fashion for the inverse depth and rotation:

$$\begin{bmatrix} \frac{1}{\hat{Z}_1} \\ \vdots \\ \frac{1}{\hat{Z}_N} \\ \hat{\omega} \end{bmatrix} = \tilde{\mathcal{C}}^\dagger \dot{\mathbf{x}} \tag{5.4}$$

[1] An instance of a spherical coordinate chart is reported in appendix 5.4.

where the symbol † denotes the pseudo-inverse. By substituting this result into equation (5.3),

$$\dot{\mathbf{x}} = \tilde{\mathcal{C}}\tilde{\mathcal{C}}^{\dagger}\dot{\mathbf{x}},$$

one ends up with an *implicit constraint* on the direction of translation, which is represented by $V(\theta, \phi)$. After rearranging the terms and writing explicitly the pseudo-inverse, one gets the following [45]:

$$\left[I - \tilde{\mathcal{C}}\left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}\tilde{\mathcal{C}}^T\right]\dot{\mathbf{x}} \doteq \tilde{\mathcal{C}}^{\perp}\dot{\mathbf{x}} = 0. \tag{5.5}$$

It is then possible to exploit this constraint for recovering the direction of translation by solving the following nonlinear optimization problem:

$$\hat{V} = \arg\min_{V \in \mathbf{S}^2} \|\tilde{\mathcal{C}}^{\perp}(\mathbf{x}, V)\dot{\mathbf{x}}\|. \tag{5.6}$$

In other words one seeks for the best vector in the two-dimensional sphere such that $\dot{\mathbf{x}}$ is the null space of the orthogonal complement of the range of $\tilde{\mathcal{C}}(\mathbf{x}, V)$. If the matrix $\tilde{\mathcal{C}}$ was invertible, the above constraint would be satisfied trivially for all directions of translation. However, when $2N > N + 3$, $\tilde{\mathcal{C}}\tilde{\mathcal{C}}^{\dagger}$ has rank at most $N + 3$, and therefore $\tilde{\mathcal{C}}^{\perp}$ is not identically zero.

Note that the solution consists in "adapting" the orthogonal complement of the linear space generated by the columns of $\tilde{\mathcal{C}}$ – which is highly structured as a function of $V(\theta, \phi)$ – until a given vector $\dot{\mathbf{x}}$ is its null space. Heeger and Jepson [45] first solved this task by minimizing the two-norm of the above constraint (5.6) using a search over θ, ϕ on a sampling of the sphere.

In section 5.2 we rephrase the Subspace constraints described in this section as a nonlinear and implicit dynamic model. Estimating motion corresponds to identifying such a model with the parameters living on a sphere: we propose a principled solution for performing the optimization task, which takes into account the temporal coherence of motion and the geometric structure of the residual (5.6).

5.1.2 Recovery of rotation and depth

Once the direction of translation has been estimated as $\hat{V} = V(\hat{\theta}, \hat{\phi})$, we may use eq. (5.4) to compute a least-squares estimate of the rotational velocity and inverse depth from

$$\begin{bmatrix} \frac{1}{Z_1} \\ \vdots \\ \frac{1}{Z_N} \\ \omega \end{bmatrix} = \tilde{\mathcal{C}}^{\dagger}(\mathbf{x}, \hat{\theta}, \hat{\phi})\dot{\mathbf{x}}. \tag{5.7}$$

Note that, from the variance/covariance of the estimation error of the direction of translation θ, ϕ, it is possible to characterize the second-order statistics of the estimate of the rotational velocity, Σ_ω. We may therefore design a simple linear Kalman filter which uses the above estimates as "pseudo-measurements" and is based upon the linear model

$$\begin{cases} \omega(t+1) = \omega(t) + n_{rw} \\ \tilde{\mathcal{C}}^{\dagger}_{2N+1:2N+3}(\mathbf{x}, \theta, \phi)\dot{\mathbf{x}} = \omega(t) + n_\omega \end{cases} \tag{5.8}$$

where the notation $\tilde{\mathcal{C}}^{\dagger}_{2N+1:2N+3}$ stands for the rows from $2N+1$ to $2N+3$ of the pseudoinverse of the matrix $\tilde{\mathcal{C}}$; n_{rw} is the noise driving the random walk model, which is to be intended as a tuning parameter, and n_ω is an error whose variance Σ_ω is inferred from the variance of the estimation error for θ, ϕ.

5.2 Solving the Subspace optimization with a dynamic filter

In this section we will view the Subspace constraint from a different perspective. Instead of considering it an algebraic set of nonlinear equations to be solved for the direction of heading, we view it as a nonlinear and implicit dynamical system, which has parameters constrained onto a two-dimensional sphere. Then we introduce a local identifier based upon an Implicit Extended Kalman Filter in order to recursively estimate the heading direction. Once the heading is estimated, it can be fed into a simple linear Kalman filter that estimates the rotational velocity.

Let us define $\alpha \doteq [\theta, \phi]^T$ as the local coordinate parametrization of the translational velocity V; θ is the

azimuth angle, and ϕ the elevation. $\mathbf{x}^i$ are measured up to some error,

$$\mathbf{y}^i \doteq \mathbf{x}^i + n^i, \tag{5.9}$$

which we model as white, Gaussian and zero-mean: $n^i \in \mathcal{N}(0, \Sigma_{n^i})$. In the presence of outliers, this hypothesis is violated, and we will show in section 5.3.3 how to detect and reject such outlier measurements before they can affect the estimation process. The error in the location of the features induces an error in the derivative,

$$\mathbf{y}'^i = \dot{\mathbf{x}}^i + n^{i'},$$

which is usually approximated by either the optical flow, or by first differences of feature positions between time t and $t+1$. Note that $\mathbf{y}'$ is an actual measurement, and is not derived from $\mathbf{y}$. This is the reason for the notation $\mathbf{y}'$ in place of $\dot{\mathbf{y}}$. Therefore, $n^{i'}$ is not the derivative of the n^i, and can actually be considered independent. Call $\mathbf{x}$ the column vector obtained by stacking the components of $\mathbf{x}^i$, similarly with $\dot{\mathbf{x}}$. Now define $\tilde{\mathcal{C}}^\perp(\mathbf{x}, \alpha)$ as in (5.3). Then the Subspace constraint (5.5) may be written as $\tilde{\mathcal{C}}^\perp(\mathbf{x}, \alpha)\dot{\mathbf{x}} = 0$. Now

$$\begin{cases} \tilde{\mathcal{C}}^\perp(\mathbf{x}, \alpha)\dot{\mathbf{x}} = 0 & V(\alpha) \in \mathbf{S}^2 \\ \mathbf{y}^i \doteq \mathbf{x}^i + n^i & \forall\, i = 1 \dots N \end{cases} \tag{5.10}$$

represents a nonlinear implicit dynamical system of a particular class, called Exterior Differential Systems [16]. *Solving for the translational velocity is equivalent to identifying the above Exterior Differential System with parameters $V(\alpha)$ on a differentiable manifold* (the sphere in this case) from the noisy data $\mathbf{y}$.

5.2.1 Identifying motion using local implicit filtering

The direction of translation, encoded by the two-dimensional vector α, is represented in the above model (5.10) as an unknown parameter which is subject to three types of constraints. First of all, $V(\alpha)$ is constrained to belong to the unit-sphere in $\mathbb{R}^3$. Secondly, the dynamics of the states $\mathbf{x}$ induces trivially a constraint on the outputs $\mathbf{y}$:

$$\tilde{\mathcal{C}}\left(\mathbf{y}, \alpha(t)\right)\mathbf{y}\prime = \tilde{n} \tag{5.11}$$

where $\tilde{n}$ is a residual noise induced by the measurement noise n. The parameters α must evolve in such a way that the outputs $\mathbf{y}$ satisfy the above dynamics. Since the outputs are directly measured, we could call the above constraint the "a-posteriori" dynamics. However, often times the direction of translation is not free to change arbitrarily, for there is some "a-priori" dynamics it must satisfy. For instance, if the camera is mounted on a vehicle, it must move according to its kinematics and dynamics, which results in a model of the generic form

$$\alpha(t+1) = f(\alpha, n_\alpha) \tag{5.12}$$

where n_α summarizes all the significant parameters of the vehicle. If the camera is hand-held, or the mechanics of its support is unknown, we know at least that velocity must be a continuous function and the acceleration cannot exceed certain values. In lack of a mechanical model, one may employ statistical models as a means of describing some inertia. For instance models of the form

$$\alpha(t+1) = f(\alpha) + n_\alpha \qquad n_\alpha \in \mathcal{N}(0, \Sigma_\alpha) \tag{5.13}$$

where f is a polynomial function and n_α is a white, zero-mean Gaussian noise.

By putting these three constraints together, we can write a discrete dynamic model for the parameters

$$\begin{cases} \alpha(t+1) = f(\alpha(t)) + n_\alpha(t) \\ \tilde{\mathcal{C}}\left(\mathbf{y}, \alpha(t)\right) \mathbf{y}\prime = \tilde{n} \end{cases} \tag{5.14}$$

$$\alpha \in [0, \pi) \times [-\frac{\pi}{2}, \frac{\pi}{2})$$

which can be used for designing an Implicit Extended Kalman filter, whose equations we report in the next subsection. Before doing that, however, we would like to stress that the function f in the model equation (5.14) is a design parameter which is left to the engineer, and depends upon the circumstances in which the algorithm is to be used.

If the algorithm is intended for general purposes, one may choose a *conservative* model, which is a model that fits a larger class than the actual one, neglecting more specific dynamics that may be present, for instance, in vehicle guidance, helicopter flight etc. . Should further information about the dynamics of the support of the camera be available, it can easily be exploited by inserting it into the model (5.13).

A typical case in which no model like (5.13) can be found is when there is no temporal coherence between subsequent images, which are snapshots of a scene taken from various points of view at different time instants. In such a case, a batch method is most appropriate. Since we are interested in real-time estimation, we always assume that the images are taken sequentially from a camera, so that temporal coherence between subsequent images is guaranteed.

In this chapter, we consider the very simplest instance of a statistical model, which is a first-order random walk:

$$f(\alpha) = \alpha. \tag{5.15}$$

It is not superfluous to point out that the first-order random walk (Brownian motion) does not restrict the motion to having constant velocity. The variance of the noise driving it, Σ_α, can be considered a tuning parameter that trades off the "speed of convergence" with the "precision" required. One may consider this as a starting point: if the dynamics of the camera in a particular experiment are not captured by this simple model, one can move up the class and consider richer models. It is our experience, however, that a first-order random walk works quite well in most cases, in the sense that it allows decent precision while not limiting the range of possible motions to a significant extent. In the experimental section we will show how the simple Brownian motion performs on a variety of situations, ranging from constant-velocity motion, to sinusoidal, to discontinuous velocity, without changing any tuning or modeling parameters.

5.2.2 Equations of the estimator

From the model (5.14), it is immediate to derive the equation for an Extended Kalman Filter (EKF) [55, 58] that estimates the direction of translation α. The only caveat is that the measurement equation is in *implicit* form. The key observation is that the vector

$$\epsilon(t) \doteq \tilde{\mathcal{C}}^{\perp}(\mathbf{y}(t), \hat{\alpha}(t+1|t))\mathbf{y}' \tag{5.16}$$

plays the role of the "pseudo-innovation" process, and therefore the standard equations of the EKF can be applied [55]. We report here the complete set of equations for the filter that estimates the direction of translation using a first-order random walk model. The reader interested in a detailed derivation of the

Implicit Extended Kalman Filter may find it in appendix F.

Prediction step

$$\begin{cases} \hat{\alpha}(t+1|t) = \hat{\alpha}(t|t) & \hat{\alpha}(0|0) = \alpha_0 \\ P(t+1|t) = P(t|t) + \Sigma_\alpha(t) & P(0|0) = P_0 \end{cases}$$

Update step

$$\begin{cases} \hat{\alpha}(t+1|t+1) = \hat{\alpha}(t+1|t) + \\ \quad + L(t+1)\tilde{\mathcal{C}}^\perp(\mathbf{y}(t), \hat{\alpha}(t+1|t))\mathbf{y}' \\ P(t+1|t+1) = \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + \\ \quad + L(t+1)D(t+1)\Sigma_{\bar{n}}(t+1)D^T(t+1)L^T(t+1) \end{cases}$$

where

$$\begin{cases} L(t+1) = P(t+1|t)C^T(t+1)\Lambda^{-1}(t+1) \\ \Lambda(t+1) = C(t+1)P(t+1|t)C^T(t+1) + \\ \quad + D(t+1)\Sigma_{\bar{n}}(t+1)D^T(t+1) \\ \Gamma(t+1) = I - L(t+1)C(t+1) \\ D(t+1) \doteq \left(\frac{\partial \tilde{\mathcal{C}}^\perp \dot{\mathbf{x}}}{\partial[\mathbf{x}(t), \dot{\mathbf{x}}]}\right)_{|[\mathbf{y}(t), \mathbf{y}'], \hat{\alpha}(t)} \\ C(t+1) \doteq \left(\frac{\partial \tilde{\mathcal{C}}^\perp \dot{\mathbf{x}}}{\partial \alpha(t)}\right)_{|\mathbf{y}(t), \hat{\alpha}(t)} \end{cases}$$

and $\Sigma_{\bar{n}}$ is the variance/covariance matrix of the measurement error $\bar{n} \doteq [n, n']$, considered as a white noise[2]. Σ_α is a tuning parameter that corresponds to the variance of the noise driving the random walk model.

At each step, the estimates of the direction of translation can be used for *instantaneously* recovering the rotational velocity from (5.7). Such a pseudo-measurement may also be used for updating the state of a linear Kalman filter based upon the model (5.8):

Prediction step

$$\begin{cases} \hat{\omega}(t+1|t) = \hat{\omega}(t|t) & \hat{\omega}(0|0) = \omega_0 \\ P_\omega(t+1|t) = P_\omega(t|t) + \Sigma_{rw}(t) & P_\omega(0|0) = P_{\omega_0} \end{cases}$$

[2]It should be noted that $\bar{n}$ is *not* a white noise, for n and n' are effectively correlated. A technique for fixing this inconvenient is described in appendix F. However, we find that the performance achieved by approximating $\bar{n}$ with a white noise is satisfactory in most cases.

Update step

$$\begin{cases} \hat{\omega}(t+1|t+1) = \hat{\omega}(t+1|t) + \\ \quad + L_\omega(t+1)\left(\tilde{C}^{\dagger}_{2N+1:2N+3}(\mathbf{y},\hat{\alpha})\mathbf{y}\prime - \hat{\omega}(t+1|t)\right) \\ P_\omega(t+1|t+1) = \Gamma_\omega(t+1)P_\omega(t+1|t)\Gamma^T_\omega(t+1) + \\ \quad + L_\omega(t+1)\Sigma_\omega(t+1)L^T_\omega(t+1) \end{cases}$$

where the gain matrices L_ω, Γ_ω are the usual ones of the linear Kalman Filter [58].

It is easy to verify that both the models (5.14) and (5.8) are locally-weakly observable. In fact, the uniqueness results in the analysis of the algorithm of Jepson and Heeger [56] are equivalent to the assessment of the observability of the model (5.14), for it is instantaneously observable. The model (5.8) is observable, for the state and measurement models are the identity and the filter just acts as a smoother. Note that the algorithm just presented produces a measure of the reliability of the estimates in the form of the second order statistics of the estimation error P and P_ω.

5.3 Implementation and experimental assessment

5.3.1 Enforcing rigid motion: the positive depth constraint

When estimating motion from visible points, we must enforce the fact that the measured points are *in front of the viewer*. This may be easily done in the prediction step by computing the mean distance of the centroid and checking whether it is positive. If it is negative, the antipodal point of the state-space sphere is chosen as the prediction.

When we do not impose such a constraint, the filter may converge to a rigid motion which corresponds to points moving behind the viewer, and is therefore not physically realizable. However, if we allow such a condition to happen by releasing the positive depth constraint, and then feed the estimate into a structure estimation, such as for example a simple Extended Kalman Filter [78, 84, 93] initialized with points at positive depth and a large model-error variance, the result is a *rubbery percept of structure* which has been observed also in psychophysical experiments [62]. A pictorial representation of the rubbery percept is illustrated in figure 5.1.

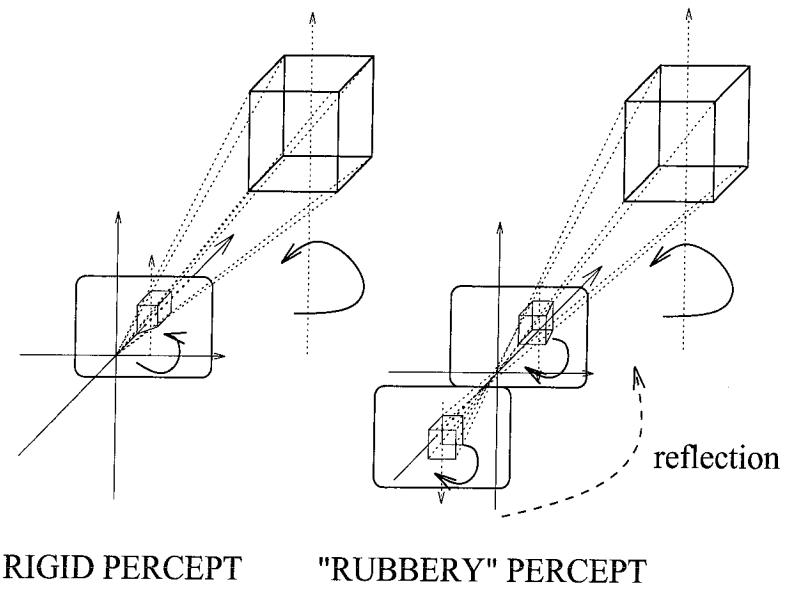

Figure 5.1: *Pictorial illustration of the "rubbery" perception: motion is estimated without imposing the positive depth constraint; this may result in a motion estimate which is compatible with a rigid structure behind the viewer. Once such a structure is interpreted as being in front of the viewer, it gives rise to the perception of a "rubbery" structure rotating in the opposite direction.*

5.3.2 Independence from structure estimation

It is worth noticing that the state of the filter proposed contains only the motion parameters, and is therefore independent from the structure of the observed scene, provided some general-position conditions. Such conditions are satisfied when the scene cannot be embedded in a planar surface, and the motion relative to the viewer generates non-zero parallax. Such conditions describe a zero-measure set in the possible structure and motion configurations, and the noise in the image-plane coordinates is sufficient to set the model in general position. As a consequence, we do not need to track a specific set of features; instead, at each step we can change set of features or locations where we compute the optical flow/feature tracking, without causing discontinuities in the estimates of motion. This is a key property of the filter, since it allows us to deal easily with occlusion and appearance of new features.

Also, note that the filter is able to work properly even when the number of visible features drops down to less than five (for small accelerations), since it integrates over time the information from each incoming frame. This, together with the robustness and noise-rejection properties, is a substantial advantage over two-views schemes.

5.3.3 Outlier rejection

One of the crucial features of the Subspace filter, as well as the Essential filter, is its independence from the structure of the scene. However, each feature-point is indirectly represented via the innovation process (5.16). In particular, for each feature-point with projective coordinates $\mathbf{x}_i$, the components of the innovation ϵ_i, defined in (5.16), describe how such a feature-point is compatible with the current estimate of motion $\hat{\alpha}$. Since at each step the filter computes the pseudo-innovation vector, it is possible to compare each component against the same at the previous time instant and, using some simple statistics, reject the measurements that give too large a residual before updating the estimates of motion. This technique may be applied both for rejecting outliers, such as mismatches in the optical flow, T-junctions, specularities etc. and for segmenting the scene into a number of independently moving rigid objects, as in chapter 8.

5.3.4 Implementation

We have implemented the filter using `Matlab`. Each update step consists essentially in 15 products of matrices of size varying from 2×2 to $2N \times 2N$, one inversion of the $2N \times 2N$ variance of the pseudo-innovation, 5 sums and the computation of the Singular Value Decomposition (SVD) of $\tilde{C}$, for a total of circa 1 Mflop for $N = 20$ points. However, the computation can be cut in half by taking into account the sparse structure of the matrices involved in the computation (block-diagonal structure of Σ_n and $\tilde{C}$). A time-consuming part of the algorithm is also the linearization of the system with respect to the measurements, $D(t+1)$.

Since the Extended Kalman Filter is based upon the assumption that the linearization error is negligible, which is not often the case, we have added to the variance $D\Sigma_{\bar{n}}D^T$ a small symmetric random matrix in order to account for the linearization error. This practice typically improves the performance of the Extended Kalman Filter for models which are strongly nonlinear.

A crucial part of the design of an EKF consist in "tuning" it, i.e. in assigning a value to the elements of the variance/covariance matrices of the model errors: $\Sigma_\alpha, \Sigma_{rw}$. A custom procedure is to assume that these matrices are diagonal, and then play with their values until the prediction error is as white as possible. Standard tests are available for this procedure, such as the "cumulative periodogram" (the integral spectrum of the prediction error). In our experiments we have performed a coarse tuning by changing the variances of the model errors by one order of magnitude at a time. We did not perform any ad-hoc or fine tuning, and

the setting was the same throughout the different experiments.

In all experiments, unless stated otherwise, the filter was initialized to zero: $\alpha_0 = 0, \omega_0 = 0$, and the initial variance of the estimation error P and P_ω was the identity matrix of dimension 2 and 3 respectively, scaled by 100.

In order to implement the filter, the linearization of the model is needed. In the appendix to this chapter we report the detailed computation of the local linearization of the measurement model.

5.3.5 Scale information recovery

The scheme proposed recovers the direction of translation as a normalized vector of $\mathbb{R}^3$. Such a normalization is necessary because of the presence of a global scale-factor ambiguity that affects the norm of translation and the inverse depth of the visible features, as it can be seen from the equation (5.1). The important fact to realize is that there is only *one* scalar ambiguity for the whole sequence so that, should some scale information become available at any instant, it can be propagated across time and the scale ambiguity resolved.

In fact, at each step the normalized translation and the rotational velocity estimated by the filter may be used for computing some "normalized" structure, which can be re-sized to fit the scale information available, as done in [93]. If no scale information is available, the initial translation may be used as a unit scale, or the distance between any two features, for instance.

5.3.6 Simulation experiments

We have generated at random a set of 20 points in space, distributed uniformly in a cubic volume of side 1 m, with the centroid placed 1.5 m ahead of the image plane. The points are projected onto an image plane of 512×512 pixels with focal length of 750 pixels. The cloud of points rotates about its centroid with a velocity of circa 5^o/frame, with the centroid maintained on the optical axis at a fixed distance from the center of projection; White, zero-mean Gaussian noise is added to the projections. The motion is roto-translational in the viewer's reference frame, and is challenging since the effects of rotation and translation superimpose.

Convergence is reached from *zero initial conditions* and noise in the image plane coordinates up to 8 pixel std. The convergence of the main filter with a noise level of 1 pixel std is reported in figure 5.2, while the same experiment is repeated with a noise level of 8 pixels std in figure 5.3. In both cases the positive depth

constraint has been enforced. The transient for converging from zero initial conditions ranges from 5 to 40 steps, depending on the noise level, the type of motion and the structure of the scene.

The least-squares pseudo-measurements of the rotational velocity, computed as described in section 5.1.2, are plotted in figure 5.4 (dashed lines), and compared with the recursive estimates (solid line) using the linear Kalman Filter described in section 5.1.2 with a noise level of 1 pixel std.

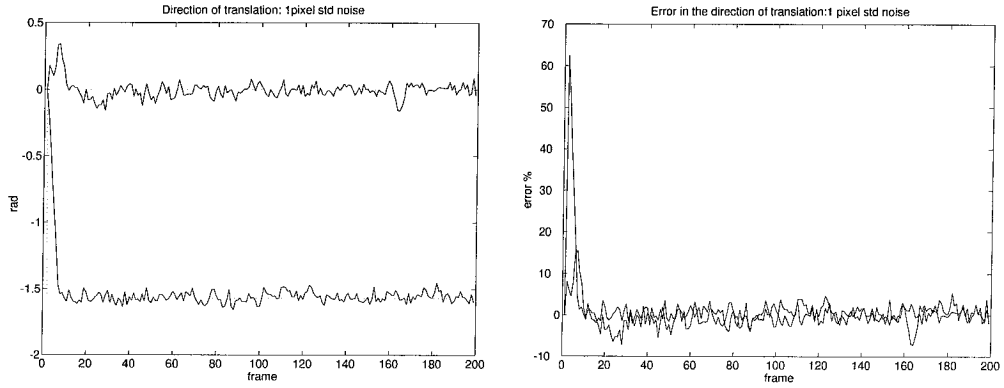

Figure 5.2: *Estimates and errors for the direction of translation when the noise in the image plane has a standard deviation of 1 pixel (according to the performance of common optical flow/feature tracking schemes). Ground truth is displayed in dotted lines. In the left plot the elevation angle ϕ is constant and equal to zero, the azimuth θ is close to $-\frac{\pi}{2}$. Note that convergence is reached from zero initial conditions in about 10 steps.*

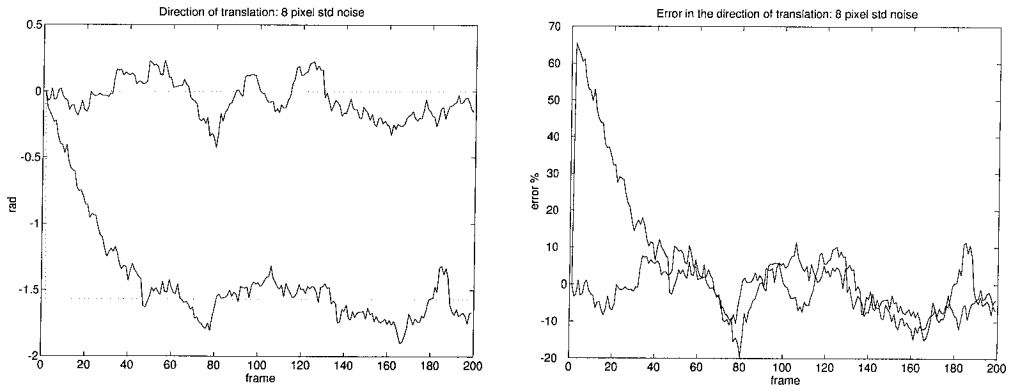

Figure 5.3: *(Left) Estimates of the two components of the direction of translation. In the left plot the elevation angle ϕ is constant and equal to zero, the azimuth θ is close to $-\frac{\pi}{2}$. The noise in the image plane measurements had 8 pixel standard deviation. The initial conditions were zero for both components. The ground truth is in dotted lines. (Right) Estimation error for the direction of translation. With noise of 8 pixel std in the data, the estimates are still within 20 % of the true value.*

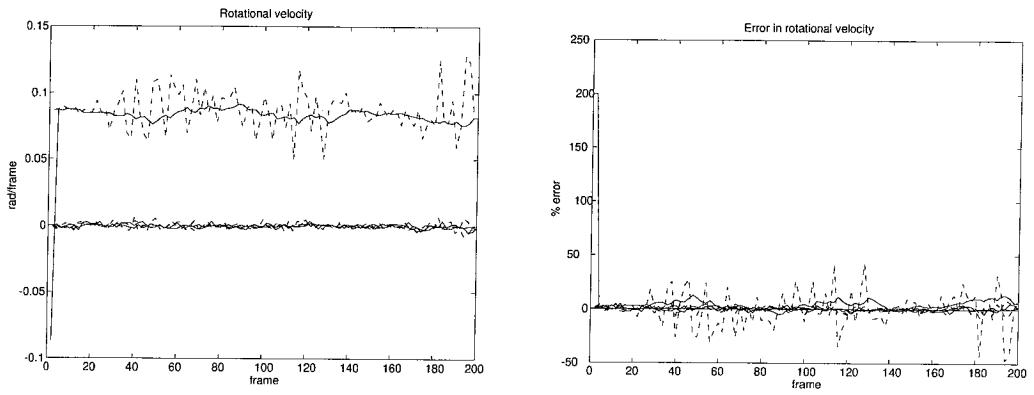

Figure 5.4: *Estimates for the components of rotational velocity (left) and corresponding error (right). Ground truth is displayed in dotted lines; the filtered estimates are in solid lines. The least-squares computation of the rotational velocity is in dashed lines.*

Challenging the model

In designing the estimator of the parameters α for the model (5.14), we have wide open choice on the dynamical model for the state f, depending upon the conditions in which the algorithm is applied. For instance, if the camera is mounted on a mobile vehicle, we may use the kinematics and dynamics of the support for describing the evolution of the state. If we know that the camera is moving with considerable inertia, we may employ a smoothness constraint etc. . In lack of any model, we can employ statistical models, for example fixed order random walks. In the experiments reported here we have chosen the simplest possible, which is the first order, corresponding to a Brownian motion. Whether this model is rich enough to capture the possible motions undergone by the camera is a question of modeling which is left to the engineer, who has to judge the intrinsic tradeoff between flexibility (large model variance) and accuracy or "smoothness" (small model variance).

Just for the sake of illustration, we have considered the same synthetic experiment described in the previous section, and modulated the speed of rotation about the object's axis first with a *sinusoid*, then with a *saw-tooth* discontinuous function, and then with a *second order* random walk (which is one step up the ladder of the class of random walks, and cannot be captured in principle by the Brownian motion). During the latter phase we have also altered the other components of the rotational and translational velocity. Eventually, motion resumed to constant velocity. Note that the parameter which is modulated is the most difficult to estimate, since the effects of rotation and translation are similar (it is one of the manifestations

of the so-called "bas-relief ambiguity"). In order to appreciate the precision of the tracking, we have lowered the noise level down to a tenth of a pixel. In figure 5.5 we show the three components of the rotational velocity (solid lines) superimposed to the ground truth (dotted lines). The two spherical coordinates of the direction of translation are plotted in figure 5.6 (solid lines) along with the ground truth (dotted lines). The estimates of the filter follow closely the motion parameters, even at the discontinuities. It is worth pointing out that the tuning was exactly the same in all the experiments in this paper, and no ad-hoc tuning was performed. It is possible to see a small, but not zero-mean, estimation error, which is a clear symptom that the model employed (a first order random walk) does not capture the true dynamics of the parameters (sinusoidal, discontinuous or a second-order random walk). If one wanted to get rid of these effects, a higher-order random walk should be considered. However, the one just performed is an extreme experiment, and usually real sequences taken from video exhibit a considerable amount of inertia. Therefore we will restrict ourselves to the simplest first-order random walk.

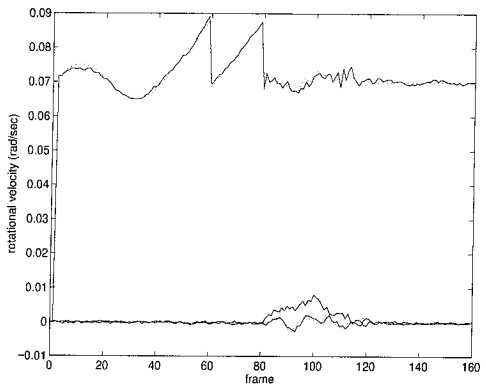

Figure 5.5: *Convergence of the filter with a first-order random walk state model in the presence of non-smooth parameter dynamics. The components of the rotational velocity of the camera are first modulated by a sinusoidal, then by a discontinuous saw-tooth and then they drift with a second order random walk before returning to the initial constant-velocity setting. The estimates (solid lines) follow the ground truth (dotted lines) despite it evolves according to dynamics which are not captured by the state model of the filter.*

The residual plot in the state-space

A typical plot of the residual function, which is the value of the Subspace constraint (5.16) as a function of the parameters $\theta \in [0, \pi), \phi \in [-\frac{\pi}{2}, \frac{\pi}{2})$, is shown in figure 5.7 for a particular value of the states. The residual depends both on the motion and structure parameters. For an isotropic cloud of dots undergoing

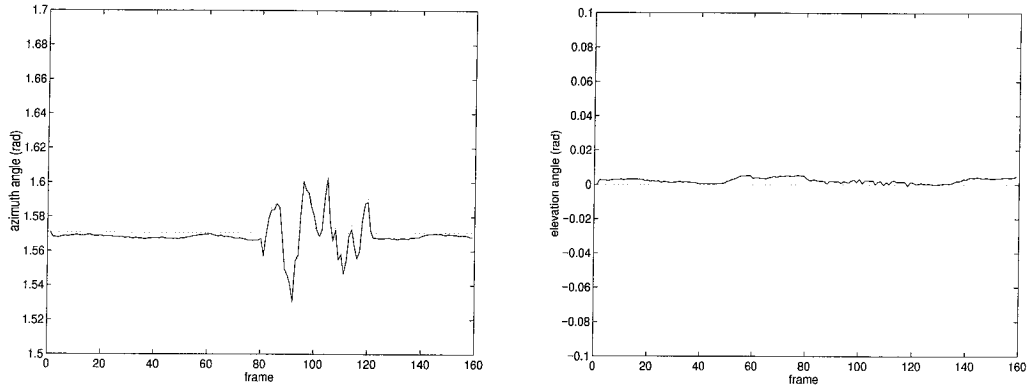

Figure 5.6: *Spherical components of the translational velocity for the experiment with non-constant velocity: azimuth (left) and elevation (right). While the rotational velocity is modulated with sinusoids and saw-tooths, translation is held constant. Between frames 80 and 120 the parameters drift according to a second-order random walk. It can be noticed that the filter follows the estimates with a small but non-zero-mean estimation error. This is due to the fact that the model that generates the data is not captured by the model used for the estimation.*

constant-velocity motion, the residual is nearly constant. Therefore, it is sufficient to show just one frame of the residual with the filter trajectory superimposed. In the following subsections we restrict our attention to the constant-velocity case just because – the residual function being constant – it is possible to display it. The bright areas indicate a small residual value. The black asterisk indicates the motion (in the local coordinates of the sphere of directions of translation) which generated the residual. It is noted that the minimum of the residual is displaced from the true motion when the norm of the rotational velocity is large. This is due to the fact that we approximate the velocity of the projected points (motion field) with first differences; the approximation is good as long as $R \doteq e^{\omega\wedge} \cong I + \omega\wedge$, i.e. as long as the norm of the rotational velocity is small.

Convergence and local minima

The reader may have noticed the presence of local minima in the plots of the residual function (figures 5.7-5.11): if motion is estimated *instantaneously* from two frames, as in [45], the estimate can be trapped into a local minimum. In our experiments, however, we have rarely witnessed convergence to a local minimum, unless temporary. This is due to the recursive nature of the scheme, which integrates information over a large baseline. In figure 5.9 and 5.10 we show a typical example of the temporary convergence of the filter to a local minimum: after few iterations the observations are no longer compatible with the motion

interpretation, forcing the filter out of the local minimum.

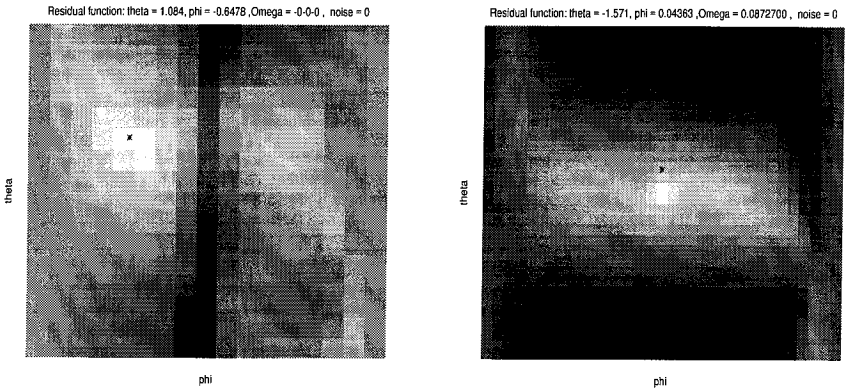

Figure 5.7: *Brightness plots of the residual function. The value of the residual is plotted on the state-space of the filter, which are the local coordinates of the sphere of directions of translation. Bright regions denote small residuals. The black asterisk is the "true" motion which generated the residual. Note that for small rotations (left) the minimum of the residual coincides with the true motion. When the rotational velocity is large (right) the Euler step approximation is no longer valid, and the minimum moves from the true location.*

Rubbery motion

A qualitatively different local minimum is the one corresponding to the "rubbery motion". When the positive depth constraint is not enforced the filter may converge either to the rigid or to the rubbery interpretation (figure 5.8). In figures 5.9 and 5.10 (left) we show the convergence to the "rubbery motion interpretation" when the positive depth constraint is released.

In figures 5.9 and 5.10 (right) we show the convergence of the filter to the rigid interpretation. Note that, when the positive depth constraint is enforced, the estimate is reflected onto the correct rigid interpretation (figure 5.11).

Structure estimation

When we feed the motion estimates into a structure-from-motion module initialized with points at positive depth and a large model-error variance [93], we may observe either a rigid set of points which move according to the correct motion (a top view of the points is shown in figure 5.12 left) or to a "rubbery" percept (figure 5.12 right). This is in accordance with the experience in psychophysical experiments [62]. Note that the rubbery solution disappears as soon as we impose the positive depth constraint.

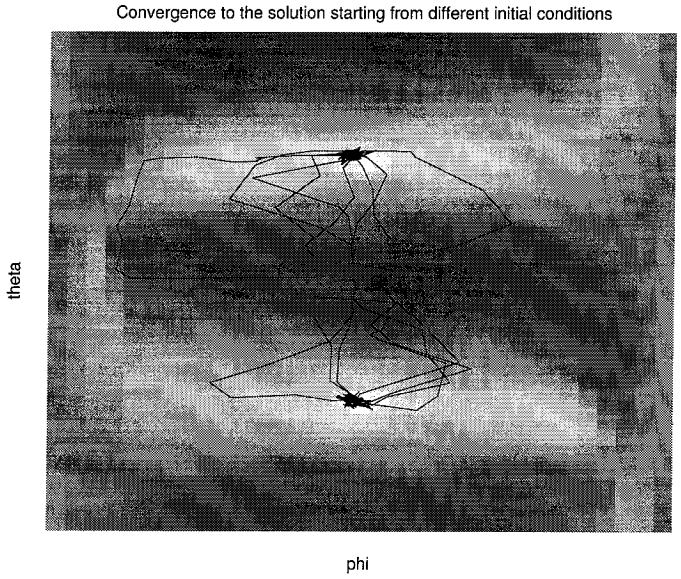

Figure 5.8: *Convergence when the positive depth constraint is not imposed and the initial condition is chosen at random around the origin (which appears in the center of the plot): a number of trajectories is shown in black solid lines superimposed on the brightness plot of the residual function. The filter may converge to either the correct rigid interpretation (bright region on the top half of the plot) or to the local minimum corresponding the "rubbery" interpretation (bright area on the bottom half of the plot).*

5.3.7 Experiments with real image sequences

The "Beckman corridor" sequence

The complete "Beckman corridor" sequence consists of a sequence of approximately 8000 frames taken by J.-Y. Bouguet et al. inside the corridor of the Beckman Institute at the California Institute of Technology. On the walls sheets of paper with high contrast provide sufficient texture for point-feature tracking. The sequence is taken while the camera moves along the corridor on top of a cart which is hand-pushed following a prescribed path on the floor of the corridor, so that qualitative ground-truth can be reconstructed. The sequence, with the tracking of about 400 feature-points, the same employed in [12], has been kindly provided to us by J.-Y. Bouguet. The features come with a condition number that indicates the presence of sufficient contrast along both spatial directions.

We show here only the first 1800 frames, during which the cart was turning of 90 degrees at a corridor angle, and then following a shallow s-turn. The algorithm makes no assumption about the fact that motion occurs on a plane, so that we can check whether the rotation about the fronto-parallel axis and the cyclo-

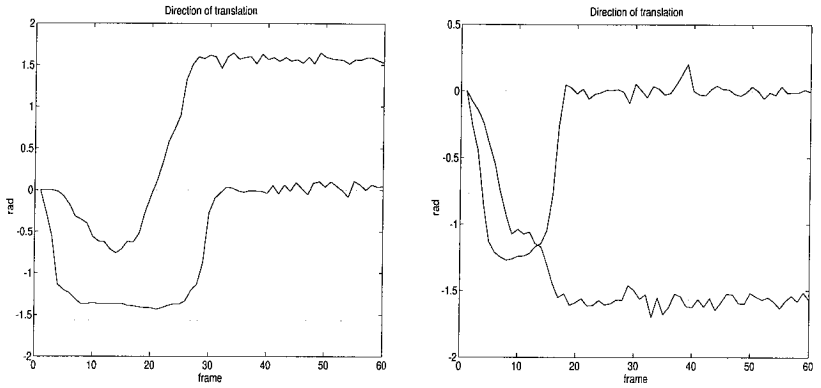

Figure 5.9: *(Left) convergence to a shallow local minimum and then to the local minimum corresponding to the rubbery interpretation when the positive depth constraint is not enforced. (Right) convergence to a shallow local minimum and then to the correct rigid motion (see also figure 16).*

rotation are estimated as zero, and the elevation angle is constant. Rotation about the vertical axis should integrate at about 90 degrees at the end of the experiment.

We have run our algorithm by using only part of the feature-set. We have fixed the maximum number of features to 20, so that the average number that pass the innovation test described in section 5.3.3 is about 15, with a minimum of 3 features at frame 400. The number of features used by the algorithm as a function of the current frame is plotted in figure 5.16. It must be noticed that no particular attention is paid to the location in the image-plane of the features used by the algorithm, so it can happen that at some step the scheme uses few features that cover only a small portion of the visual field.

In figure 5.14 we show the estimated direction of translation, consisting of the azimuth angle (direction of heading) and elevation angle. The latter is constant to about 5 degrees, which corresponds to the angle between the camera and the horizontal axis on the cart. The direction of heading points left during the first turn, then slightly right and then left again during the s-turn. This is consistent with the cart having front steering wheels and the camera being mounted on the front. The rotation angle about the Y-axis (horizontal) and Z-axis (cyclo-rotation) are zero, as reported in figure 5.15. The rotational velocity about the vertical axis X, reported in figure 5.16, shows first the full left turn, then the s-turn left-right. The integral of the velocity along the whole sequence is 101^o, with an overall error of about 10^o over 1800 frames. This is the mean integral of the error along the whole sequence. In order to appreciate the convergence of the filter, which was initialized to zero, we show the components of the main filter for the direction of heading, along

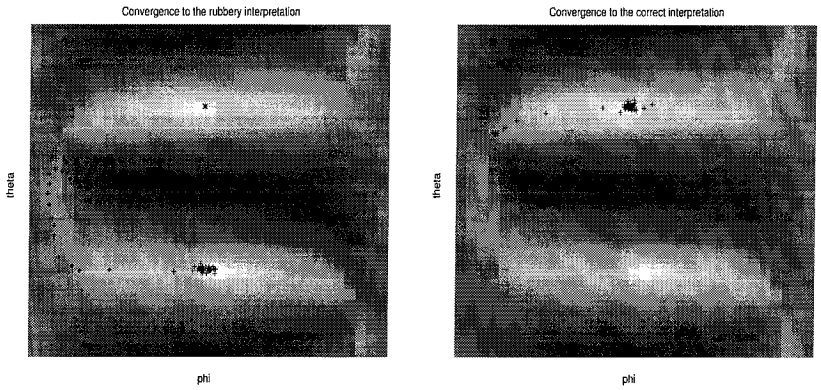

Figure 5.10: *Convergence to the "rubbery interpretation" (left) versus convergence to the rigid motion interpretation (right). The state of the filter at each step is represented as a black '+' and superimposed to the average residual function (darker tones for larger residuals). After the transient, the states accumulate either around the local minimum corresponding to the rubbery interpretation (the one on the bottom half of the plot) or to the one corresponding to the true motion, on the upper half of the plot. The trajectory of the state is also plotted component-wise in figure 15.*

with the variance of the estimation error – plotted as errorbars – during the first 100 frames (figure 5.17).

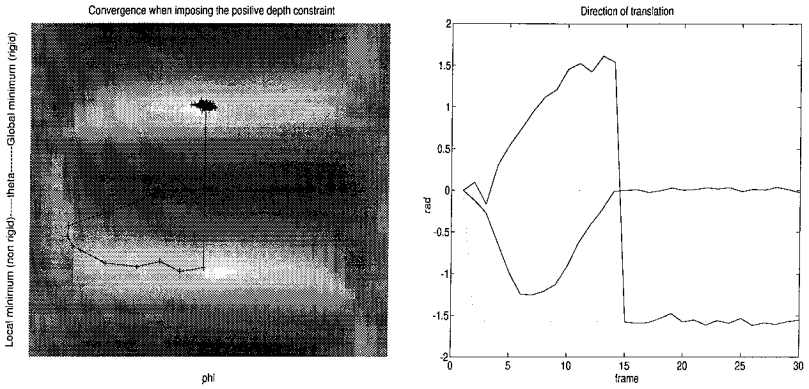

Figure 5.11: *Convergence when the positive depth constraint is enforced: (left) trajectory of the filter on top of the brightness plot of the residual function, (right) corresponding motion components. Initial conditions are zero.*

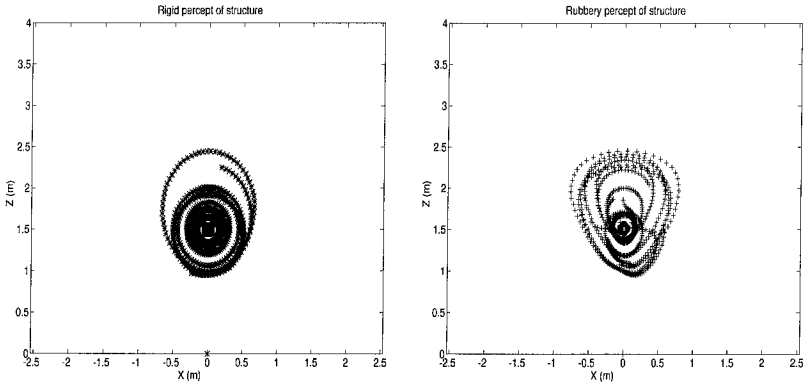

Figure 5.12: *Convergence of a structure-from-motion module to a rigid interpretation of structure (left) or to a rubbery object rotating in the opposite direction (right). The plots show a top view of the points, with the image plane on the lower end.*

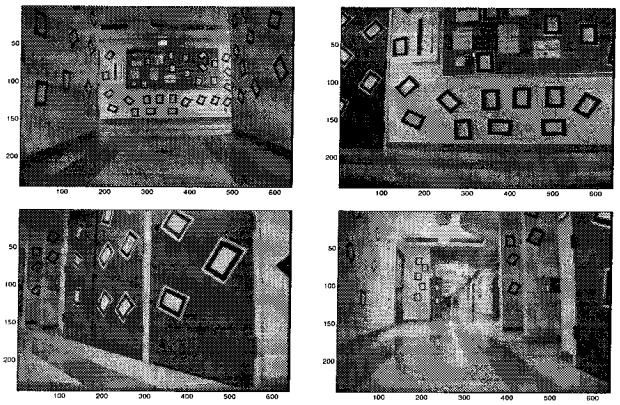

Figure 5.13: *Few images from the "Beckman sequence". The camera is mounted on a cart which is pushed around a corridor. First the cart turns left by 90^o, then right and left again on a s-turn. The sequence consists of approximately 8000 frames. We have processed here only the first turn of the corridor, which corresponds to the first 1800 frames. The sequence was taken by Bouguet et al., who also performed the feature tracking using Sum of Square Differences criteria on a multi-scale framework.*

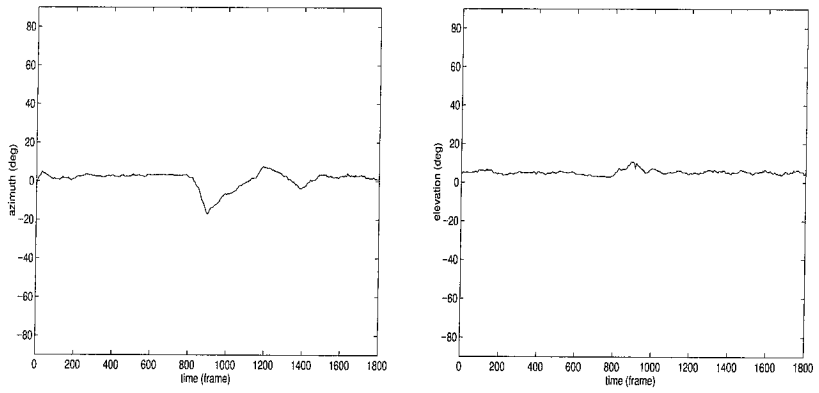

Figure 5.14: *(Left) Azimuth angle for the corridor sequence. Zero corresponds to forward translation along the Z-axis. The first peak is due to the left turn, while the subsequent wiggle corresponds to a right-left s-turn. (Right) Elevation angle. The camera was pointing downwards at an angle of approximately 5^o; therefore the heading direction was approximately constant with an elevation of $+5^o$. Since the camera was hand-held, there is quite a bit of wobbling.*

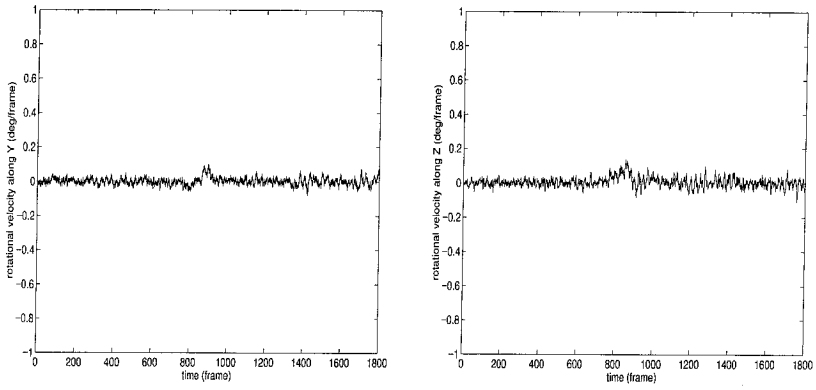

Figure 5.15: *Rotational velocity about the Y-axis (left) and about the Z-axis (right). Since the camera was not pitching nor cyclo-rotating, both estimates are close to zero as expected. Since the camera was hand-held and no accurate ground-truth is available, it is not easy to sort out the effects of noise and the ones of small motions or vibrations of the camera.*

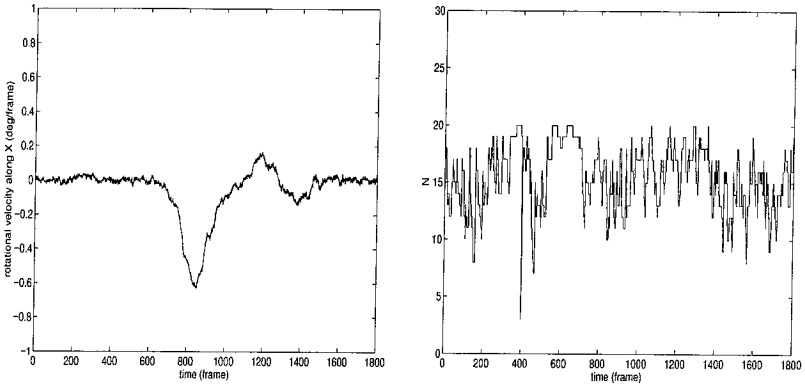

Figure 5.16: *(Left) Rotational velocity about the vertical axis. First the camera turns left at the corner of the corridor (frames 700 to 1000), then right and then left again around the s-turn (frames 1000 to 1600). The integral of the rotational velocity should add up to approximately* 90^o*, for this is the change of orientation of the camera from beginning to end. The sum of the estimates is* 101^o*, corresponding to an error of* 10% *circa on a sequence of 1800 frames. (Right) Number of features employed by the algorithm at each time step. On average the algorithm uses 15 feature-points, without particular attention to how they are distributed on the image plane. The maximum number of features used is 20, and the minimum is 3. Note that two-frames algorithms would not perform in such a case, since at least 5 features need to be visible at all times. The temporal integration involved in the filter, on the contrary, allows us to retain the estimates even in presence of less than 5 features.*

Appendices

5.4 Computation of the local linearization of the Subspace model

In this appendix we give the detailed equations for the linearization of the model of the Subspace filter. We compute the derivative of the implicit measurement equation

$$\tilde{\mathcal{C}}^{\perp}(\mathbf{x}, V(\theta,\phi))\dot{\mathbf{x}} \tag{5.17}$$

as a function of the derivative of $\tilde{\mathcal{C}}$ with respect to the states θ, ϕ and the measurements $\mathbf{x}$. From the definition of $\tilde{\mathcal{C}}^{\perp}$ we have

$$\tilde{\mathcal{C}}^{\perp} \doteq \left(I - \tilde{\mathcal{C}}\left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}\tilde{\mathcal{C}}^T\right). \tag{5.18}$$

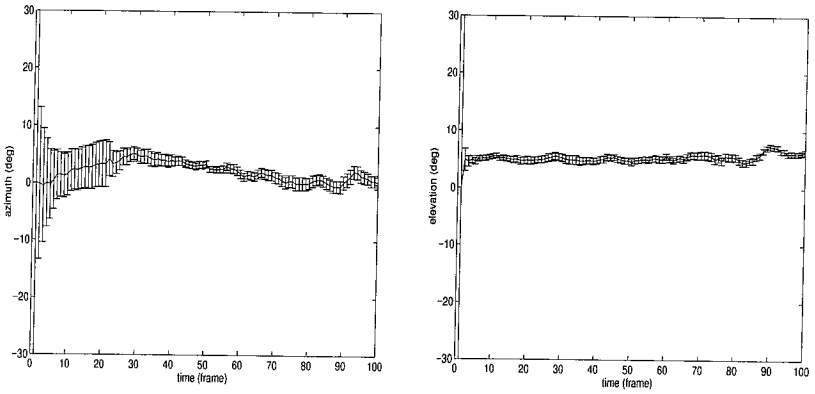

Figure 5.17: *Close-up view of the transient in the estimates of the direction of translation (azimuth on the left, elevation on the right). The variance of the estimation error, represented using the error-bars, decreases during the first 20-30 frames, after which it remains bounded around the current estimate of the parameter.*

If we call α a scalar parameter (α will be either $\phi(t), \theta(t)$ or one component of the measurements $x^i(t)$, $y^i(t)$) and

$$\tilde{\mathcal{C}}_\alpha \doteq \frac{\partial \tilde{\mathcal{C}}}{\partial \alpha} \tag{5.19}$$

then we have

$$\begin{aligned} \tilde{\mathcal{C}}_\alpha^\perp = & - \tilde{\mathcal{C}}_\alpha \left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}\tilde{\mathcal{C}}^T - \tilde{\mathcal{C}}\left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}\tilde{\mathcal{C}}_\alpha^T - \\ & - \tilde{\mathcal{C}}\frac{\partial\left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}}{\partial\alpha}\tilde{\mathcal{C}}^T. \end{aligned} \tag{5.20}$$

Since, for a square and invertible matrix A, $A_\alpha^{-1} = -A^{-1}A_\alpha A^{-1}$, we have

$$\begin{aligned} \tilde{\mathcal{C}}_\alpha^\perp = & -\tilde{\mathcal{C}}_\alpha \left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}\tilde{\mathcal{C}}^T - \tilde{\mathcal{C}}\left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}\tilde{\mathcal{C}}_\alpha^T - \\ & - \tilde{\mathcal{C}}\left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}\left(\tilde{\mathcal{C}}_\alpha^T\tilde{\mathcal{C}} + \tilde{\mathcal{C}}^T\tilde{\mathcal{C}}_\alpha\right)\left(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}}\right)^{-1}\tilde{\mathcal{C}}^T \end{aligned} \tag{5.21}$$

we can write, after collecting the common terms,

$$\tilde{\mathcal{C}}_\alpha^\perp = -\tilde{\mathcal{C}}^\perp\tilde{\mathcal{C}}_\alpha\tilde{\mathcal{C}}^\dagger - \tilde{\mathcal{C}}^{\dagger^T}\tilde{\mathcal{C}}_\alpha^T\tilde{\mathcal{C}}^\perp. \tag{5.22}$$

If we call

$$\mathcal{K}_\alpha \doteq \tilde{\mathcal{C}}^\perp \tilde{\mathcal{C}}_\alpha \tilde{\mathcal{C}}^\dagger \tag{5.23}$$

and we notice that $\tilde{\mathcal{C}}^\perp$ is a symmetric matrix, we end up finally with

$$\tilde{\mathcal{C}}_\alpha^\perp = -\mathcal{K}_\alpha - \mathcal{K}_\alpha^T. \tag{5.24}$$

We now seek for a cheaper and better-conditioned way of computing the matrix $\mathcal{K}$. Consider the Singular Value Decomposition of the matrix $\tilde{\mathcal{C}}$:

$$\tilde{\mathcal{C}} = U_c \Sigma_c V_c^T \tag{5.25}$$

then it is immediate to notice that

$$\tilde{\mathcal{C}}^\perp = I - U_c U_c^T. \tag{5.26}$$

After substituting for the SVD of $\tilde{\mathcal{C}}$ and exploiting the orthogonality of U and V, we have

$$\mathcal{K}_\alpha = \left(I - U_c U_c^T\right) \tilde{\mathcal{C}}_\alpha V_c \Sigma_c^{-1} U_c^T. \tag{5.27}$$

In order to compute the full linearization of the implicit measurement equation with respect to the states θ, ϕ and the measurements $\mathbf{x}$, we are only left with computing the derivatives of the matrix $\tilde{\mathcal{C}}$ with respect to these parameters:

$$\tilde{\mathcal{C}}_\theta = \begin{bmatrix} \mathcal{A}_1 \frac{\partial V}{\partial \theta} & & & 0 \\ & \ddots & & 0 \\ & & \mathcal{A}_N \frac{\partial V}{\partial \theta} & 0 \end{bmatrix} \tag{5.28}$$

$$\tilde{\mathcal{C}}_\phi = \begin{bmatrix} \mathcal{A}_1 \frac{\partial V}{\partial \phi} & & & 0 \\ & \ddots & & 0 \\ & & \mathcal{A}_N \frac{\partial V}{\partial \phi} & 0 \end{bmatrix} \tag{5.29}$$

$$\tilde{\mathcal{C}}_{x^i} = \begin{bmatrix} \ddots & & & 0 \\ & \begin{bmatrix} V_3 \\ 0 \end{bmatrix} & & \frac{\partial \mathcal{B}_i}{\partial x^i} \\ & & \ddots & 0 \end{bmatrix} \tag{5.30}$$

$$\tilde{\mathcal{C}}_{y^i} = \begin{bmatrix} \ddots & & & 0 \\ & \begin{bmatrix} 0 \\ V_3 \end{bmatrix} & & \frac{\partial \mathcal{B}_i}{\partial y^i} \\ & & \ddots & 0 \end{bmatrix} \tag{5.31}$$

where

$$\frac{\partial V}{\partial \theta} = \begin{bmatrix} -cos(\phi)sin(\theta) \\ cos(\phi)cos(\theta) \\ 0 \end{bmatrix} \tag{5.32}$$

$$\frac{\partial V}{\partial \phi} = \begin{bmatrix} -sin(\phi)cos(\theta) \\ -sin(\phi)sin(\theta) \\ cos(\phi) \end{bmatrix}. \tag{5.33}$$

The spherical coordinates are defined such that

$$V(\theta, \phi) \doteq \begin{bmatrix} cos(\theta)cos(\phi) \\ sin(\theta)cos(\phi) \\ sin(\phi) \end{bmatrix}. \tag{5.34}$$

We now have all the ingredients necessary for computing the linearization of the model:

$$C \doteq \left(\frac{\partial \tilde{\mathcal{C}}^{\perp} \dot{\mathbf{x}}}{\partial [\theta \ \phi]} \right) = \begin{bmatrix} \tilde{\mathcal{C}}^{\perp}_{\theta} \dot{\mathbf{x}} & \tilde{\mathcal{C}}^{\perp}_{\phi} \dot{\mathbf{x}} \end{bmatrix} \tag{5.35}$$

$$D \doteq \left(\frac{\partial \tilde{\mathcal{C}}^{\perp} \dot{\mathbf{x}}}{\partial [\mathbf{x} \ \dot{\mathbf{x}}]} \right) = \begin{bmatrix} \tilde{\mathcal{C}}^{\perp}_{x^1} \dot{\mathbf{x}} & \tilde{\mathcal{C}}^{\perp}_{y^1} \dot{\mathbf{x}} & \dots & \tilde{\mathcal{C}}^{\perp}_{y^N} \dot{\mathbf{x}} & | & \tilde{\mathcal{C}}^{\perp} \end{bmatrix}. \tag{5.36}$$

Chapter 6 Weak perspective and the bas-relief ambiguity

Weak-perspective is a scaled orthographic projection that can be used for approximating the imaging model when the field of view is small as well as the scene "flat" relative to its distance from the viewer. A well-known visual effect under such conditions is the so-called "bas-relief ambiguity": the rotational velocity along an axis parallel to the image-plane and the depth parameters are hard to observe.

In previous chapters we have seen how reduction can be useful to remove structure parameters and therefore obtain smaller models of constant dimension in spite of occlusions. In this chapter we want to further decouple the space of motion parameters into a portion that is not affected by the bas-relief ambiguity and one which contains parameters subject to such an ambiguity. By doing so, we can obtain consistent estimates of the states that are not affected by the bas-relief ambiguity even when the remaining states are impossible to recover.

Background and notation

We consider a number N of point-features P^i of coordinates $\mathbf{X}^i \in \mathbb{R}^3$ $\forall i = 1 : N$, that project perspectively onto the image plane in the points $p^i \in \mathbb{R}P^2$. The weak-perspective points

$$p^i \doteq \pi(P^i), \text{ of coordinates } \mathbf{x}^i \doteq \frac{\begin{bmatrix} \mathbf{X}^i_1 \\ \mathbf{X}^i_2 \end{bmatrix}}{\bar{d}}, \tag{6.1}$$

where $\bar{d}$ is the average distance between the scene and the center or projection, can be considered an approximation to the true projection under a small visual angle and a negligible relief. If the scene undergoes a rigid motion $g \in SE(3)$ – which can be represented as an instantaneous translation $T \in \mathbb{R}^3$ and a rotation matrix $R \in SO(3)$ – then the motion of the points in 3-D and the weak-perspective measurements describe a nonlinear dynamical system of the form

$$\begin{cases} P^i(t+1) = g_t \circ P^i(t) \in \mathbb{R}^3 & \textit{rigid motion} \\ g_{t+1} = g_t \oplus n_g(t) \in SE(3) & \textit{small acceleration} \\ p^i(t) = \pi(P^i(t)) \in \mathbb{R}^2 & \textit{weak} - \textit{perspective} \end{cases} \tag{6.2}$$

where $\oplus$ represents a first order random walk in the local coordinates of the motion parameters (as a simple mean of modeling some inertia). In the case of constant-velocity we simply have $n_g = e \in SE(3)$.

Outline of the chapter

In principle, a state observer for the dynamical model described above could be employed for estimating jointly the structure and rigid motion of the scene from weak-perspective. Such an observer might, however, exhibit poor performance due to the presence of structure in the state, which results in a state-space that has high and changeable dimension, as points get occluded or move out of the visual field. One possible strategy consists in trying to reduce the dimensions of the state-space as much as possible, ending up with a small-dimensional highly-constrained state-space model. In the next section we are going to explore the principles of reduced-order observers, which are the basis for constructing dynamic models for estimating motion independent of the scene's structure.

Then we will see how these principles can be applied to isolate the components of the motion parameters that are affected by the bas-relief ambiguity.

6.1 The general principle: pushing the reduced order observer

6.1.1 Reducing the order of the model

We start from the model (6.2) and operate a change of coordinates (into observable canonical form) in order to linearize the measurement equation, which leads us to a model in the form

$$\begin{cases} p^i(t+1) = f_1(p^i, g_t, \bar{d}) + f_2(g_t, \bar{d}) s^i \in \mathbb{R}^2 \\ s^i(t+1) = h_1(p^i, g_t, \bar{d}) + h_2(g_t, \bar{d}) s^i \in \mathbb{R} \\ g_{t+1} = g_t \oplus n_g(t) \in SE(3) \\ \mathbf{y}^i = p^i + n^i \in \mathbb{R}^2 \qquad\qquad \forall\, i \;=\; 1 \ldots N \end{cases} \tag{6.3}$$

where $s^i \doteq \mathbf{X}^i_3/\bar{d}$ is the relative depth of each point and n^i is a measurement noise which is assumed to be zero-mean, white and Gaussian. We omit the time argument when it is t.

The first step towards reducing the order of the model consists in eliminating from the state the variables that are directly measured. Using a technique extrapolated from the so-called *reduced-order observer* [57], one can "solve" the measurement equation for the states one wishes to eliminate:

$$p^i = \mathbf{y}^i - n^i \tag{6.4}$$

and then substitute them into the dynamics of the remaining states in equation (6.3):

$$\begin{cases} s^i(t+1) = h_1(\mathbf{y}^i, g_t, \bar{d}) + h_2(g_t, \bar{d})s^i + n_{s_i} \\ \bar{d}(t+1) = l(\mathbf{y}^i, g_t, s^i, \bar{d}) + n_d \\ g_{t+1} = g_t \oplus n_g(t) \qquad \in \{SE(3) \bmod \mathbb{R}\}. \end{cases} \tag{6.5}$$

The dynamics of the average depth $\bar{d}$ has been isolated from the other scaled depths s^i, since it will play a role in the coordinatization of the motion parameters in the presence of the scale-factor ambiguity (section 6.2.1). The original measurement equation becomes now trivial; however, the dynamics of the variable being eliminated becomes the new measurement constraint, which involves one time-delay:

$$\mathbf{y}^i(t+1) - f_1(\mathbf{y}^i, g_t, \bar{d}) - f_2(g_t, \bar{d})s^i = \tilde{n}^i. \tag{6.6}$$

The noise terms n_{s_i}, n_d and $\tilde{n}^i$ are induced by the measurement noise n^i. In principle, the time-delay could be eliminated from the measurement equation (6.6) using an output-dependent change of coordinates [57]. Here we do not pursue this approach, and we are content with keeping two images in memory at each time. The notation $\{SE(3) \bmod \mathbb{R}\}$ reminds us that there is an overall scale ambiguity in recovering the motion parameters; as a result, we represent g_t as a normalized translation $T \in \mathbf{S}^2$ and a rotation matrix $R \in SO(3)$.

6.1.2 Decoupling structure from motion

With the simple procedure described above, we have reduced the state-space from $3N + 5$ down to $N + 6$, while adding a time-delay to the $2N$ measurements. One could push this idea even further, and eliminate the N parameters s^i from the $2N$ new measurement equations (6.6):

$$s^i = f_2^\dagger(g_t, \bar{d}) \left(\mathbf{y}^i(t+1) - f_1(\mathbf{y}^i, g_t, \bar{d}) \right), \tag{6.7}$$

where $\dagger$ denotes the subspace inverse, and then substitute them into the dynamics of the remaining states in (6.5). The measurement equation no longer becomes trivial, for there are still N independent constraints

that have not been employed for "eliminating" s^i:

$$f_2^\perp(g_t, \bar{d}) \left(\mathbf{y}^i(t+1) - f_1 \right) = f_2(g_t, \bar{d})^\perp \tilde{n}^i \tag{6.8}$$

where $\perp$ denotes the subspace orthogonal complement. Again, the dynamics of the variable being eliminated becomes a measurement constraint, with now two delays. The final expression of the model becomes therefore of the form

$$\begin{cases} \bar{d}(t+1) = l(\mathbf{y}^i, g_t, f_2^\dagger \mathbf{y}^i(t+1) - f_2^\dagger f_1, \bar{d}) + n_d \\ g_{t+1} = g_t \oplus n_g(t) \qquad \in \{SE(3) \bmod \mathbb{R}\} \\ f_2^\perp(g_t, \bar{d}) \left(\mathbf{y}^i(t+1) - f_1 \right) = f_2(g_t, \bar{d})^\perp \tilde{n}^i \\ f_2^\dagger(g_t, l) \left(\mathbf{y}^i(t+2) - f_1(\mathbf{y}^i(t+1), g_t, l) \right) + \\ \qquad -h_1 - h_2 f_2^\dagger \left(\mathbf{y}^i(t+1) - f_1 \right) = \tilde{n}_{s_i} \end{cases} \tag{6.9}$$

where the arguments in f_1, h_1, h_2 and l have been omitted and $\tilde{n}_{s_i}$ is, as usual, a noise induced by substituting the measurements into the dynamics of s^i. We can write the above model in a more synthetic form as

$$\begin{cases} \xi(t+1) = m(\xi(t)) + n_\xi(t) \qquad \in M \sim \mathbb{R}^6 \\ \chi\left(\xi(t), \mathbf{y}(t), \mathbf{y}(t-1), \mathbf{y}(t-2)\right) = n_\chi(t) \ \in \mathbb{R}^{3N} \end{cases} \tag{6.10}$$

where ξ belongs to a six-dimensional state-space manifold that encodes the motion parameters and the average depth of the scene, and only $2N$ of the measurement constraints are independent.

In the experimental section 6.3 we will show an actual expression of the above model with an appropriate choice of coordinates, along with experiments of the performance of the filter derived from such a model on real and synthetic image sequences.

One could play the game just described over and over, and successively eliminate each state by solving from the time-evolution of the measurement equation and substituting into the state equations. This process is guaranteed to succeed as long as the dynamic model in question is locally-weakly observable [53]. In fact, the above process could be regarded as analogous to a level-wise inversion of the Observability Grammian in the linear case. Of course, the more levels are involved, the higher the number of delays that appear in the measurement equations (or the higher the number of Lie-derivatives of the measurements in the continuous-time case). In our case, two delays are sufficient, for the model is observable with two levels of bracketing.

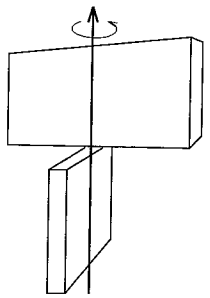

Figure 6.1: *One of the manifestations of the "bas-relief ambiguity" is evident from watching a rotating billboard. From a distance, the more slanted the surface, the faster it seem to move, while the two surfaces appear to move disjointly.*

6.2 Isolating the bas-relief ambiguity: motion decoupling and choice of coordinates

It is well-known that, under small visual angles and negligible relief, it is difficult to resolve some of the motion parameters. This effect, which is known as the "bas-relief ambiguity", can be observed for example by taking two flat surfaces, connecting them rigidly at a right angle, and rotating them about an axis (a rotating billboard, see figure 6.1). The perceived motion from a distance is strikingly non-rigid, as the surface which is more slanted seems to move faster. In the presence of the bas-relief ambiguity, it is important to parametrize the state-space manifold M so that the states that are affected are "isolated". Failure to do so may result in poor estimates of *all the states*, including the ones that are not affected by the ambiguity.

6.2.1 Choosing the motion coordinates

At this point, one may ask if it is possible to push the reasoning outlined in section 6.1.1, and formulate filters that estimates *only the motion parameters that are not affected by the bas-relief ambiguity.* The procedure outlined in the previous section relies on the fact that one is able to "solve" for the states to be eliminated from the time-evolution of the measurement equation. Eliminating p^i and s^i was easy because they appeared *linearly* in the measurement equations (6.3) and (6.6) respectively. However, it is not so for the motion parameters, which are encoded into $\xi(t)$. In this section we will see that it is possible to decouple

the motion parameters and formulate two filters, one with four states, and one with two states only, which are not affected by the bas-relief ambiguity. To this end, we need to make a choice for the local-coordinate parametrization of the motion parameters $\xi \in M \sim \mathbb{R}^6$.

We choose to represent motion using $\xi \doteq [\tilde{T}^T, \theta, \phi, \rho]^T$, defined such that

$$\tilde{T} \doteq \frac{T}{\bar{d}} \in \mathbb{R}^3 \qquad R = e^{\mathbf{e}_3 \wedge \theta} e^{[e^{\mathbf{e}_3 \wedge \phi} \mathbf{e}_1] \wedge \rho} \in SO(3) \tag{6.11}$$

where $(\mathbf{e}_3\wedge)$ is a 3×3 skew-symmetric matrix having all zeros but -1 in position $(1,2)$ and 1 in position $(2,1)$. The Euler-angle representation of rotation, which was introduced by Koenderink and Van Doorn [61], corresponds to rotating by ρ radiants about an axis on the image-plane, forming an angle ϕ with the horizontal axis, and then rotating about the optical axis by an angle θ. It has the advantage that the bas-relief ambiguity is isolated in the parameter ρ, while cyclo-rotation θ and the angle ϕ are always easy to estimate. The disadvantage is that, like all Euler-angles, it is only a local representation, and a filter based upon such a representation may run into singularities.

6.2.2 Approximate filter with four states

Under the choice of coordinates described in (6.11), eq. (6.8) may be written as

$$\begin{bmatrix} & & \vdots & \\ \mathbf{y}(t+1)^T & \mathbf{y}(t)^T & 1 & \\ & & \vdots & \end{bmatrix} \begin{bmatrix} cos(\phi)v(t) \\ sin(\phi)v(t) \\ -cos(\phi-\theta) \\ -sin(\phi-\theta) \\ -w(t) \end{bmatrix} = \tilde{n} \tag{6.12}$$

which is a rank-four homogeneous equation up to zero-mean noise. In the above equation, v and w are approximated by [107]

$$\begin{cases} v(t) \cong sin(\rho)[-sin(\phi)\ cos(\phi)]\bar{\mathbf{y}}(t) + cos(\rho) + \tilde{T}_3 \\ w(t) = sin(\rho)\left(-sin(\phi)\tilde{T}_1 + cos(\phi)\tilde{T}_2\right) \end{cases} \tag{6.13}$$

under the condition $\rho \cong 0$. Eliminating ρ from this constraint, even though simplified, is not a trivial matter. A naïf approach consists in writing a filter for the 4 variables $\psi(t) \doteq [v(t), w(t), \theta(t), \phi(t)]^T$, having (6.12)

as an implicit measurement equation. The problem is that the dynamics of ψ involves *all the states* ξ, and therefore we cannot hope to eliminate some of them and use their dynamics as a measurement equation. In fact, note that equation (6.12) comes from the residual measurement equations that were not used for eliminating s^i, but then it is necessary to integrate the measurement equations with the dynamics of the variables s^i being eliminated.

An approximate filter can be obtained, however, by modeling the dynamics of v and w as a random walk, and *neglecting the dynamics of the variables* $\tilde{T}, \rho$. Such a filter will have a reduced measurement constraint, and an approximate dynamics:

$$\begin{cases} \psi(t+1) = \psi(t) + n_\psi \ \in \mathbb{R}^4 \\ \text{eq. (6.12)} \in \mathbb{R}^N. \end{cases} \tag{6.14}$$

Note that, once the filter has estimated v and w, there is no way of unfolding the motion parameters $\tilde{T}$ and ρ out of them; we must be content with the four parameters ψ, which are only a partial representation of motion. Such an approach has been pursued, for instance, in [107], although derived differently.

6.2.3 Reduced filter with two states

The redundancy in the measurements may be exploited up to the point in which we define a filter with only two states. To this end, consider eq. (6.12), which is obtained by eliminating the relative depth parameters s^i. The motion parameters $\tilde{T}$ and ρ appear through v and w, defined in eq. (6.13). Therefore, we may eliminate these four variables and be left with a filter that has only θ and ϕ in its state. To this end, rewrite eq. (6.12) as

$$\begin{bmatrix} & & \vdots & \\ \mathbf{y}_{(t+1)}^T \begin{bmatrix} c_\phi \\ s_\phi \end{bmatrix} & & -1 & \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \mathbf{y}_{(t)}^T \begin{bmatrix} c_{\phi-\theta} \\ s_{\phi-\theta} \end{bmatrix} + \tilde{n}(t) \tag{6.15}$$

or, in a more condensed form,

$$\mathcal{G}(t+1, \phi) \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \mathcal{H}(t, \theta, \phi) + \tilde{n}(t). \tag{6.16}$$

Here c_ϕ stands for $cos(\phi)$, and so for $s_\phi \doteq sin(\phi)$. Then, eliminating $\tilde{T}$ and ρ can be easily done my eliminating v and w, which appear *linearly* in the above equation. The final expression of the model involving only ϕ and θ is therefore

$$\begin{cases} \theta(t+1) = \theta(t) + n_\theta(t) \\ \phi(t+1) = \phi(t) + n_\phi(t) \\ \mathcal{G}^\perp(t+1,\phi)\mathcal{H}(t,\theta,\phi) = n_r(t) \end{cases} \tag{6.17}$$

where n_r is the noise of the reduced constraint, which is induced by the measurement noise n, and n_θ and n_ϕ are noise models driving the random walk, whose variances are to be regarded as tuning parameters.

6.3 Experimental Assessment

We have implemented three recursive filters for the models of eq. (6.9), (6.14) and (6.17), using a local observer based upon the Implicit Extended Kalman Filter, which is derived in appendix F; the only thing needed is the model and an expression of the local linearization of the model. Space limitations do not allow us to report all of the computation; we restrict ourselves here to writing the actual equations in the local coordinates for the model (6.9) (the other two are already in local coordinates and ready for use)

$$\begin{cases} \tilde{T}(t+1) = \frac{\tilde{T}(t)}{v(t)} \\ \theta(t+1) = \theta(t) + n_\theta(t) \\ \phi(t+1) = \phi(t) + n_\phi(t) \\ \rho(t+1) = \rho(t) + n_\rho(t) \\ eq.\ (6.12) \in \mathbb{R}^N \\ \Psi \begin{bmatrix} s_\phi v(t+1)v(t) \\ -c_\phi v(t+1)v(t) \\ -s_{\phi-\theta}v(t) - s_\phi c_\rho v(t) \\ c_{\phi-\theta}v(t) + c_\phi c_\rho v(t) \\ s_\phi s_\rho^2 + s_{\phi-\theta}c_\rho \\ -c_\phi s_\rho^2 - c_{\phi-\theta}c_\rho \\ (-s_\phi \tilde{T}_1 + c_\phi \tilde{T}_2)(1-c_\rho) + \tilde{T}_3 s_\rho \end{bmatrix} = n_\Psi \end{cases} \tag{6.18}$$

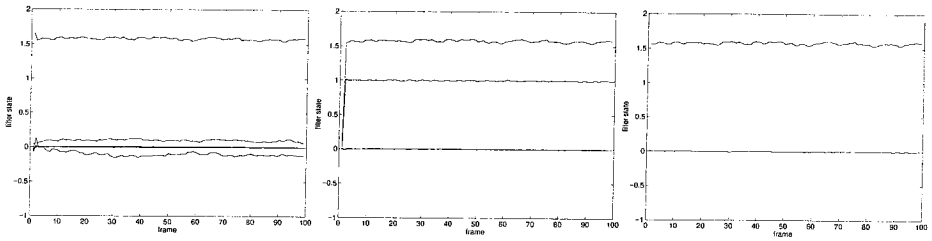

Figure 6.2: *Simulation experiment. Estimates of each filter (solid lines) along with ground truth (dotted lines) for a noise level of one tenth of a pixel std. The left plot shows the estimates of the state of the full filter with six states, the middle plot is the approximate filter with four states, and the right plot is the reduced filter with two states. Units are radiants/frame for the rotational velocity. Translation is adimensional since it is scaled to the average depth.*

where $\Psi = [\mathbf{y}^T(t+2)\ \mathbf{y}^T(t+1)\ \mathbf{y}^T(t)\ 1] \in \mathbb{R}^{N\times 7}$ and v and its dynamics are defined from equation (6.13).

All the filters have been implemented in `Matlab` and tuned with the same parameters. Tuning the filters has proven to be a rather non-trivial matter. Since the measurements are actually generated by a full-perspective projection, while the dynamic model regards them as the outcome of a scaled orthography, the variance of the measurement noise must be increased so as to account for the perspective distortion. As a consequence, the variance of the model error must be also increased in order to avoid over-smoothing. Since perspective distortion is very poorly modeled by a white and zero-mean Gaussian noise, all the filters based upon the models described above (as well as all other models based upon the weak-perspective model) perform poorly in the presence of significant perspective effects.

The three schemes have similar computational complexity and they run, in the current `Matlab` implementation, at about 2 to 10 Hertz on a Sun Sparc 20, depending upon the number of visible points (usually on the order of 10 to 100).

6.3.1 Simulation experiment

A cloud of 20 dots, 1 meter in diameter, was generated at a distance of 10 meters from a viewer and rotated about a vertical axis with a speed of about five degrees per frame. Its projection onto a virtual image plane of 500×500 pixels was corrupted with noise whose level was varying between one tenth of a pixel to one pixel std. The three filters exhibit similar performance, for the states which they have in common, in the presence of low noise levels (see figure 6.2). As the noise level increases, the full filter with 6 states is affected by the bas-relief ambiguity, so that two of its states are estimated poorly (figure 6.3). However, notice that

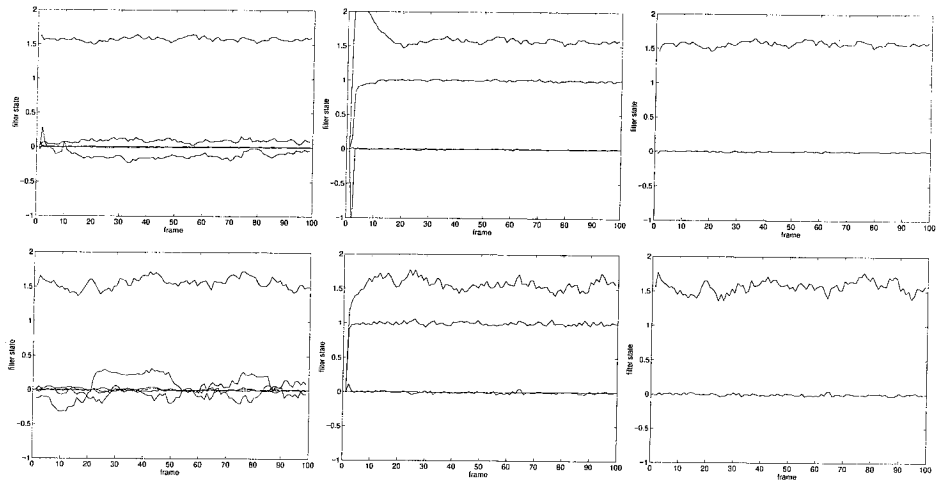

Figure 6.3: *Degradation of the estimates with increasing measurement noise. In the top row we report the behavior of the filters for a noise level of half a pixel std, and in the bottom row for one pixel std. We plot the estimates of each filter (solid lines) along with ground truth (dotted lines). The full-filter with 6 states (left column) degrades unevenly, for two of its states are subject to the bas-relief ambiguity. However, the particular choice of coordinates still allows estimating correctly the remaining 4 states which are not subject to the bas-relief ambiguity. The affine filter (central column) and reduced filter (right column) are not affected by the bas-relief ambiguity, and their estimation error increases gracefully with the increasing level of measurement noise. Units are rad/frame for the components of rotational velocity.*

the remaining states converge with estimation errors very similar to that exhibited by the approximate filter and by the reduced filter, which are not affected by the bas-relief ambiguity.

6.3.2 The arm experiment

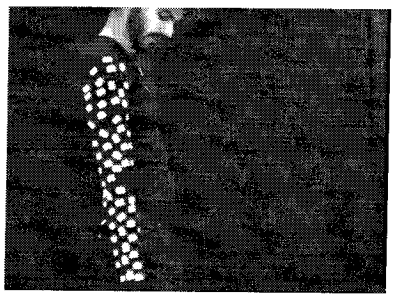

Figure 6.4: *L. Goncalves in his mimetic attire. The "arm sequence" is 250 frames long and the motion is rotatory on a plane parallel to the image plane. The arm was rotating upwards for half of the sequence, and then downwards for the rest of it.*

The "arm" experiment consists of a sequence of about 250 frames kindly provided to us by L. Goncalves. An arm with high contrast texture was rotating with a velocity of about half a degree per frame (figure

6.4). Features were selected and tracked automatically using simple gradient methods. The estimates of the full relative motion between the arm and the camera are estimated by the full filter with 6 states, as reported in figure 6.5. The estimates correspond to the qualitative ground-truth provided with the sequence. In figure 6.6, we plot the variance of each estimate represented using error-bars. Since motion is mainly cyclo-rotational, any estimate of the angle ϕ is correct. Indeed, we are in a singularity of the coordinate representation. The filter estimates ϕ as being approximately $\frac{\pi}{2}$, and correctly assigns a large variance to the estimate. The estimates of the only significant state in common among all filters are compared in figure 6.7. There we also report the cyclo-rotation as estimated by the "Susbspace filter" (as in chapter 5), which is based upon a full-perspective model. The estimates of the filters are consistent. The ones based upon the weak-perspective models are more jittery, since the variance of the measurement error has to be increased in the tuning in order to account for the perspective distortion.

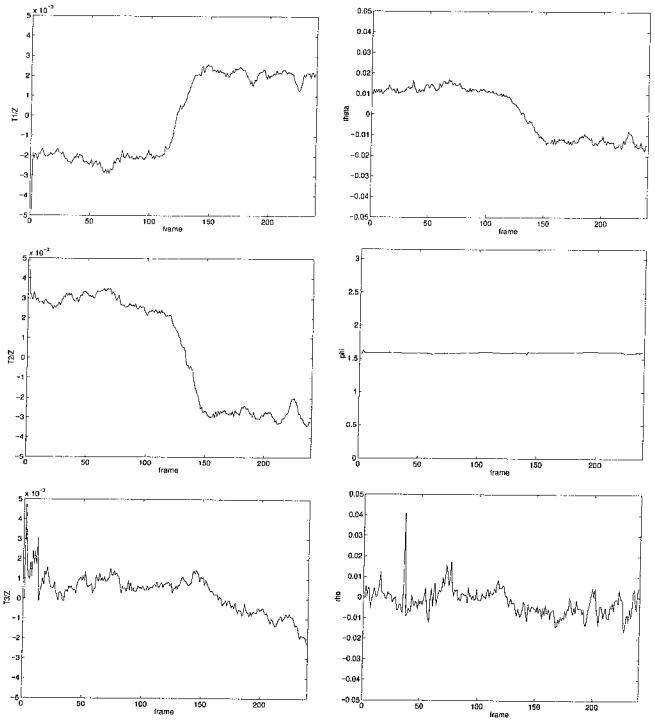

Figure 6.5: *The "arm experiment". In the left column we plot the three components of the estimated direction of translation normalized to the average depth of the scene; in the right column we display, respectively from top to bottom, the local coordinates of rotation: θ, ϕ and ρ. The algorithm was using on average 10 feature-points per frame. Units are rad/frame for the components of rotational velocity. Translation is adimensional since it is scaled to the average depth.*

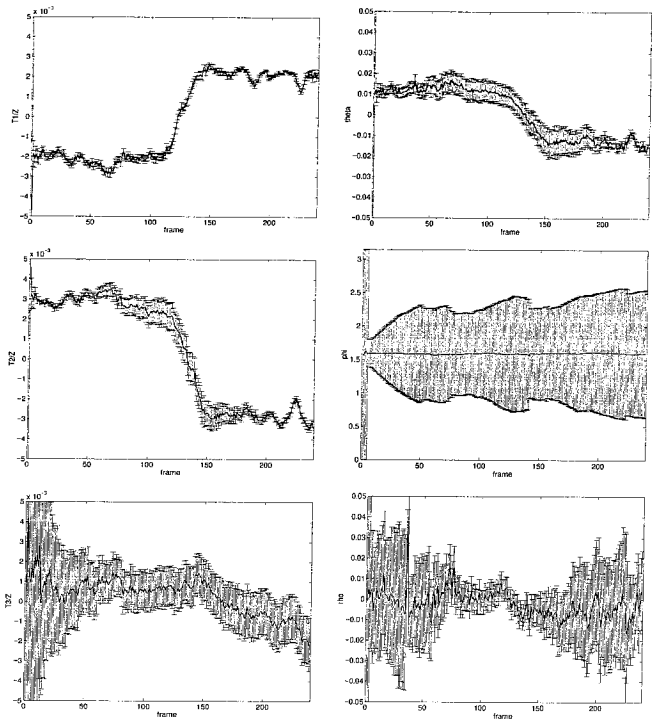

Figure 6.6: *The same estimates reported in figure are now plotted along with their variance, represented using error-bars. It can be seen that, since rotation occurs only about the optical axis, the direction of the rotation axis on the image-plane, ϕ is arbitrary, and is indeed estimated with a very large variance (middle-right plot).*

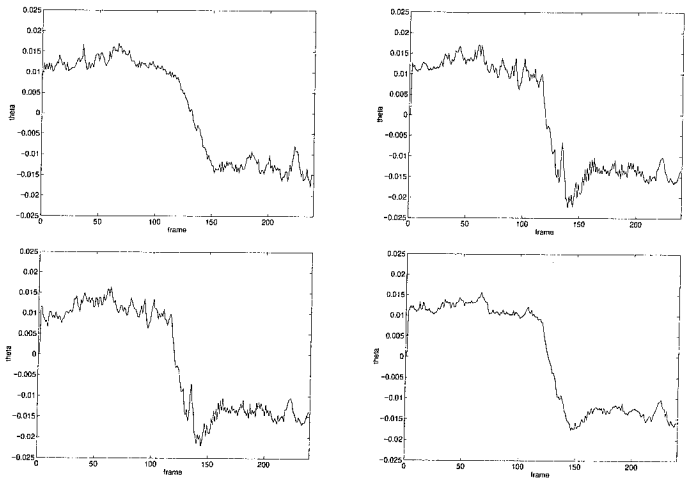

Figure 6.7: *Comparison of the estimates of the angle θ for, respectively from top to bottom, the full filter (six states), the approximate filter (four states), the reduced filter (two states), and the Subspace filter based upon full-perspective.*

Chapter 7 Pushing the reduced-order observer: fixation

In chapter 3 we have seen how it is possible to reduce the order of the models described in chapter 2 so as to eliminate structure and be left with a model for estimating motion independent of structure. In chapter 5 it was possible to further eliminate the rotational component in the continuous-time framework by eliminating it from the measurement equation and its derivatives. This is not possible *in closed form* in discrete-time case, since the rotation parameters appear at the exponential. In chapter 6 we have further eliminated states that are affected by the bas-relief ambiguity.

In this chapter we explore an alternative strategy for reducing the order of the observer. Rather than "explicitly" eliminating states, we can impose that some ("implicit") function of the state-space is constant (or "fixated"). By doing so, we impose constraints on the state-space manifold, and therefore reduce its effective dimension. The resulting models are exactly in the form of Essential models (chapter 3), where the parameters are constrained on different subspaces of the Essential manifold, depending upon which constraints are imposed.

Background and notation

In this chapter we adopt the same notation used in chapter 3.

Outline of the chapter

We first describe how forcing a function of the state-space to be constant can be used to reduce the dimension of the state-space. Since such a function is arbitrary, we choose to treat three simple examples, which correspond to imposing that the image of a point, a line or a plane are fixed.

7.1 Output stabilization and geometric stratification

Suppose that we are told that some of the states of a dynamical model are zero. Then we may as well constrain the observer to the remaining states, and eliminate the constant ones from the dynamical model.

The same applies if a *function* of the states is held constant. In fact, consider a point in the state-space manifold, $\mathbf{P} \in M$. If $f : M \to \mathbb{R}$ is a smooth function, and $0 = f(\mathbf{P})$ is a regular value, then the pre-image $f^{-1}(0) \subset M$ is a submanifold of M [41], and the point $\mathbf{P}$ is constrained onto such a submanifold. In this case it is possible to find a set of coordinates where some of the parameters are constant, and we can therefore concentrate our attention on the remaining ones.

Therefore, if we view some function of the state as an *output* (measurement equation) of the dynamic system, and this output is held constant, or *stabilized*, we may identify a "slice" of the state-manifold, and constrain the model on such a slice.

Although the choice of which function to stabilize is arbitrary, we will consider three simple instances: the image-motion of a point, a point and a line, and a plane. By stabilizing such outputs, we identify slices of the Essential manifold, which build a geometric stratification of the problem of estimating motion under fixation constraints.

7.2 Choosing a control action

In order to stabilize a particular function of the image, we could either actuate the camera, and move it in space ("mechanical control"), or pre-process the image by considering changes of coordinates that depend upon the outputs, without acting on the support of the camera ("software control"). For instance, keeping a single feature point fixed on the image plane can be accomplished both by rotating the camera about the center of projection (or about another point in space), or by shifting the origin of the image-coordinates. As far as the effects on motion estimation are concerned, the two methods are equivalent. A few gaze-control techniques which guarantee exponential convergence are described in [91], while image-shift registration techniques that achieve fixation in a single step are described, for instance, in [96].

Fixating a point and a line on the image plane may be easily achieved by fixating a point and then rotating the image until another point comes to the desired line. This may be accomplished both by rotating the camera about the fixation axis, or by rotating the image about the optical center with a purely software operation.

Fixating a plane in the image, however, can be only accomplished by manipulating, or pre-processing, the image, as described in section 7.5.1.

Stabilized feature	Compensating 3-D motion	Corresponding image deformation	Residual DOFs	State-space manifold
none	none	none	5	E Essential mfd
point	2-D camera rotation	image center displacement	4	$\mathcal{S}^4$ Sylvester mfd
point+line	rotation about optical center	image center shift + rotation	3	$\mathcal{S}^3$ 3-dimensional Sylvester mfd
plane	no feasible 3-D rigid motion	planar warping	2	$so(3)$ skew-symmetric unit-norm 3×3 matrices

7.3 Stabilization of a point (fixation)

Let us assume that we have applied any fixation technique that provides us with a sequence of images where the projection of a given point remains fixed on the image-plane. Since the projection of the fixation point is stationary, the object (scene) is free only to rotate about this point, and to translate along the fixation line. Therefore there are overall 4 degrees of freedom left from the fixation loop. These four degrees of freedom are encoded into the rotation matrix $R = e^{\Omega\wedge}$, and in the relative translation along the fixation axis $v \in \mathbb{R}$. The epipolar representation presented in chapter 3 applies immediately once we represent the translation T as

$$T(R, v) \doteq \begin{bmatrix} -R_{13} & -R_{23} & -R_{33} + v \end{bmatrix}^T, \tag{7.1}$$

and $v \doteq \frac{d(t+1)}{d(t)} \neq 0$ is the ratio between the distance of the fixation point at time $t+1$ and the same distance at time t.

The coplanarity constraint (3.9) also holds in the case of fixation, once we have substituted the appropriate expression for T. Since there are four degrees of freedom, the parameters Ω and v will now lie on a four-dimensional subspace of the Essential manifold. Indeed, it can be easily verified that the Essential matrices under the fixation constraint are all and only the 3×3 Essential matrices that satisfy the following Sylvester's equation

$$\mathbf{Q}(R, v) = RS^T + vSR \tag{7.2}$$

where $S \doteq [0\ 0\ \alpha]^T\wedge$ and α is the arbitrary scaling factor due to the homogeneous nature of the coplanarity constraint. We will call $\mathcal{S}^4$ the four-dimensional submanifold of the Essential manifold which is defined by the above equation after normalization. The $\mathcal{S}^4$ manifold is locally diffeomorphic to $\mathbb{R} \times SO(3)$ and hence to $\mathbb{R}^4$.

Therefore, in order to estimate motion under the fixation constraint, it is sufficient to consider the epipolar constraint where now the parameters are constrained not on the Essential manifold, but on the $\mathcal{S}^4$-manifold. We have therefore to deal with a model of the form

$$\begin{cases} (\mathbf{Q}\mathbf{x}^i(t))^T\mathbf{x}^i(t+1) = 0 \\ \mathbf{y}^i(t) = \mathbf{x}^i(t) + n_i(t) \end{cases} \qquad \mathbf{Q} \in \mathcal{S}^4 \tag{7.3}$$

where

$$\begin{aligned} \mathcal{S}^4 \;=\; & \{\mathbf{Q} \in E \mid \mathbf{Q} = RS^T + vSR, R \in SO(3), \\ & v \in \mathbb{R}, S = [0\ 0\ 1]^T\wedge\}. \end{aligned} \tag{7.4}$$

Estimating motion reduces to identifying the above dynamical system with parameters on $\mathcal{S}^4$.

7.4 Stabilization of a point and a line

Suppose now that, in addition to fixating a point, we can maintain a line passing through it fixed in the image plane. We are essentially in the same situation described in the previous section, once we have "frozen" the degree of freedom corresponding to cyclorotation (rotation about the optical axis). Therefore there are overall 3 degrees of freedom. The Essential matrices corresponding to motions that obey the "point plus line" fixation constraint must lie on a three-dimensional submanifold of the submanifold $\mathcal{S}^4$ of the Essential manifold E, since the point-fixation constraint described in the previous section is satisfied. The only modification that occurs is that now there is no cyclorotation. Therefore the parameter space becomes

$$\mathcal{S}^3 = \mathcal{S}^4 \cap \{R = e^{\begin{bmatrix} \omega_1 & \omega_2 & 0 \end{bmatrix}^T\wedge}\}. \tag{7.5}$$

Hence, under the "point plus line" fixation assumption, we end up with a model of the form

$$\begin{cases} (\mathbf{Q}\mathbf{x}^i(t))^T\mathbf{x}^i(t+1) = 0 \\ \mathbf{y}^i(t) = \mathbf{x}^i(t) + n_i(t) \end{cases} \qquad \mathbf{Q} \in \mathcal{S}^3 \tag{7.6}$$

which needs to be identified in order to estimate the motion parameters.

7.5 Stabilization of a plane

We now proceed in our stratification by assuming that we are able to "compensate" the image sequence in such a way that the points that lie on some plane (not necessarily a physical plane in the scene) remains fixed in the image plane. In this case there is no physical motion of the camera that achieves this compensation (besides locking the camera to the plane). Therefore we need to "deform" the images of the sequence in order to account for the motion of the plane.

7.5.1 Compensation of plane-motion: warping

Let us assume, for the moment, that all points in the scene lie on a plane – not passing through the origin – described by $\Pi = \{\mathbf{X}_\pi \in \mathbb{R}^3 \mid \mathbf{a}^T\mathbf{X}_\pi = 1\}$. We indicate with $\mathbf{x}_\pi \in \mathbb{RP}^2$ the projective coordinates of the generic point of the plane Π. We will now see that, as the plane Π moves rigidly in space, its image deforms according to a projective transformation, i.e. a linear transformation of the projective coordinates. In fact, we may write the evolution of the 3-D points of the plane as

$$\mathbf{X}^i_\pi(t+1) = R(t)\mathbf{X}^i_\pi(t) + T(t)\mathbf{a}^T\mathbf{X}^i_\pi(t) \doteq A(t)\mathbf{X}^i_\pi(t) \tag{7.7}$$

where $A(t) = R(t) + T(t)\mathbf{a}^T$ is a 3×3 invertible matrix. The projective coordinates of the points on the plane obey a similar relation

$$\mathbf{x}^i_\pi(t+1) \sim A(t)\mathbf{x}^i_\pi(t) \tag{7.8}$$

where the symbol $\sim$ indicates equality up to a scaling factor (projective equivalence). Given 4 or more point-correspondences on the image-plane, we may solve the above equation for the 8 parameters of A that are free after normalization.

Once the matrix A has been estimated, up to a scaling factor, we may *undo* the transformation by multiplying the transformed points by A^{-1}:

$$\mathbf{x}^i_\pi(t+1)^w \doteq A^{-1}\mathbf{x}^i_\pi(t+1) = \mathbf{x}^i_\pi(t). \tag{7.9}$$

Therefore, such a *warping* leaves the points of the plane fixed in the image [8, 4, 85].

7.5.2 Plane-plus-parallax representation

In the previous subsection, we have assumed that all points of the scene lie on the plane Π.

Now, let us assume that we have compensated for some plane, for instance the average plane, and see what happens to the points $\mathbf{X}^i$ that do not lie on such a plane, after the warping with A^{-1}. In general, $\mathbf{x}^i(t+1)^w \neq \mathbf{x}^i(t)$. More specifically, we have

$$\begin{aligned} \mathbf{x}^i(t+1)^w &\sim A^{-1}\mathbf{x}^i(t+1) = (R+T\mathbf{a}^T)^{-1}\mathbf{x}^i(t+1) \\ &\sim (I - R^T T\mathbf{a}^T)^{-1}R^T[R\mathbf{X}^i(t)+T] \end{aligned} \tag{7.10}$$

where $[\cdot]$ denotes the projective coordinates. If we call $T' \doteq R^T T$, then we can write

$$\begin{aligned} \mathbf{x}^i(t+1)^w &\sim (I - T'\mathbf{a}^T)^{-1}[\mathbf{X}^i(t)+T'] \\ &\sim \left(I + \frac{T'\mathbf{a}^T}{1-\mathbf{a}^T T'}\right)[\mathbf{X}^i(t)+T'] \end{aligned} \tag{7.11}$$

which may be finally written as

$$\mathbf{x}^i(t+1)^w \sim \mathbf{x}^i(t) + \beta^i(t)T' \tag{7.12}$$

where $\beta^i(t) = \left(1 + \frac{T'\mathbf{a}^T\mathbf{X}^i(t)}{1-\mathbf{a}^T T'}\right)$ is a scalar factor. Therefore, the last term can be interpreted as a residual, which is in the direction of the epipole (the projective coordinates of the direction of translation T'). The derivation above is taken from [85].

This representation, consisting in the motion of a plane – encoded by the matrix A – and the residual parallax in the direction of the epipole – encoded by $\beta^i(t)$ – is known in the literature as the "plane-plus-parallax" representation, and has been developed in [8, 4, 85].

Now, let us see how warping affects the setup of epipolar geometry. It is immediate to verify that

$$\mathbf{x}^{i^w}(t+1)^T(T'\wedge)\mathbf{x}^i(t) = 0 \qquad T' \in \mathbf{S}^2 \tag{7.13}$$

and, therefore, the effect of rotation has been canceled out by the image warping. We may represent the

overall model as, again, an implicit dynamical system, with parameters on a manifold

$$\begin{cases} (\mathbf{Q}\mathbf{x}^i(t))^T \mathbf{x}^{iw}(t+1) = 0 \\ \mathbf{y}^i(t) = \mathbf{x}^i(t) + n_i(t) \end{cases} \qquad \mathbf{Q} = T'\wedge \in so(3) \cap \mathbf{S}^2 \equiv \mathbf{S}^2 \tag{7.14}$$

where the last equivalence follows from the isomorphism between $so(3)$ and $\mathbb{R}^3$ [11]. Thus, the plane-fixation constraint corresponds to Essential matrices which are of the form $\mathbf{Q} = T'\wedge$. Due to the normalization constraint on T', we have only two degrees of freedom left, and rotation has been fully decoupled from translation. This model may be considered the discrete-time equivalent of the Subspace constraint, for it fully decouples structure and rotation, and leaves a dynamic constraint only in the direction of translation.

This argument closes the picture on model reduction in Structure From Motion, and links together chapters 2, 3, 5 and 6.

Chapter 8 Outlier rejection and segmentation

In chapters 2, 3, 5, 6 and 7 we have assumed that the scene being viewed consists of a single rigid object. The process of dividing the scene into portions that corresponds to objects moving independently is called *segmentation.* In chapter A we have also observed that, since feature tracking is an intrinsically *local* procedure, often times it may result in outlier data due to mismatch or violations of the brightness constancy equation. The process of outlier rejection may also be thought of as a segmentation of the scene, since outliers appear to move in a way which is inconsistent with the other points belonging to the same object. In this chapter we address the issue of segmentation and outlier rejection.

Many cues may be used for scene segmentation, such as boundaries, texture, discontinuities of the optical flow, stereo, motion etc. . Ultimately a system for performing three dimensional scene segmentation ought to integrate all the information available by exploiting each cue.

There are two *motion cues* that might be used for scene segmentation: 2D motion on the image plane, where optical flow discontinuities are projections of scene depth and/or 3D motion discontinuities, or 3D motion itself. There are a number of assumptions as well: object rigidity, piecewise smoothness of the scene, object opaqueness (which, together with all previous assumptions translates into piecewise smoothness of the optical flow), existence of a "dominant motion". Accordingly, the motion-based segmentation algorithms may be classified into a number of categories. 2D optical flow region-based algorithms [9, 99, 14, 52], 3D region-based [25, 97], and *transparent* 3D motion [111, 13, 87]. We call "transparent 3D motion" algorithms the ones which do not make use of regions-contiguity assumptions, and may therefore handle motion of transparent objects.

In this chapter we present a method for segmenting a scene from a sequence of monocular images using only 3D motion cues. We make no use of spatial contiguity, and hence we are able to perform on transparent motions. The main assumption is that each object populating the scene is a rigid body. We use the "Essential filter" as a motion estimator, although all the considerations apply to the other models described as well.

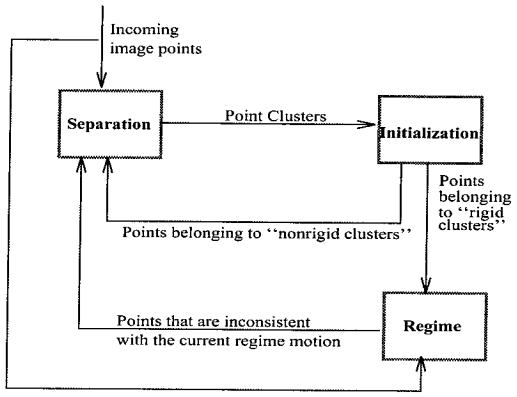

Figure 8.1: *Structure of the segmentation scheme.*

Background and notation

In this chapter we adopt the notation used to derive the "Essential filter" in chapter 3. We use the Essential filter as a testbed for simplicity, although all considerations can be extended to each of the models described in chapters 2, 5, 6, 7 as well.

Outline of the chapter

We will first sketch an outline of the algorithm. It consists of a separation step, which composes clusters of points having high probability of belonging to the same rigid object, an initialization step in which a filter is assigned to each cluster, and then a regime phase, which is characterized by having a filter associated to each rigid object. During the regime phase the rigidity assumption is constantly checked and, if the object splits into more than one independent body, the points which are incompatible with the current motion are rejected and returned to the separation phase (see fig. 8.1). In the later sections the operation of each step is analyzed in detail. In section 8.1 we introduce an innovation-based self-validating test for rejecting outliers. In section 8.2 the operation of the separation and initialization phases is explained and some open issues are discussed.

The scheme which we propose consists of three "modes of operation" which are constantly active during the segmentation procedure. A supervision program is in charge of assigning to each feature point a mode of operation (see figure 8.1).

Separation Suppose we are at the initial time instant. We do not know how many objects are moving in the scene and which points belong to which objects. The separation step produces a set of clusters (one for each point) which have high probability of belonging to a single rigid motion.

Initialization The initialization mode takes the output of the separation step, namely a set of clusters of points, and runs a motion estimation algorithm (for instance the Essential filter) in parallel for each cluster. After a settling time it gives either a convergence verdict, which promotes the cluster to the regime stage, or a divergence verdict, which causes the cluster to be assigned to the separation again.

Regime The clusters which are promoted from the initialization mode enter into "regime" mode. Each object is assigned to a filter which is in charge of estimating the rigid motion of the object and constantly checking for outliers (points whose motion is not consistent with the rigid interpretation). This is done using a very simple criterion which we call the "predicted innovation test"[1].

[1] This choice may sound like an oximoron, since the innovation is exactly what cannot be predicted. However, it renders the idea of what the test is about.

8.1 The innovation as a residual

The Essential model presented in chapter 3 may be written as

$$\begin{cases} \mathbf{x}^{i^T}(t+1)\mathbf{Q}(T(t),R(t))\mathbf{x}^i(t) = 0 \\ \tilde{\mathbf{x}}^i(t) = \mathbf{x}^i(t) + n^i(t) \end{cases} \quad \forall i = 1:N \tag{8.1}$$

where $\mathbf{x}^i(t)$ are the projective coordinates of each of the N visible points in the viewer's reference at time t, $\mathbf{Q} \doteq R(T\wedge)$, where (T,R) is the rigid motion undergone by the observer between time t and $t+1$ and $\tilde{\mathbf{x}}^i$ are the noisy measurements of the image plane coordinates. It is customary to assume $n^i \in \mathcal{N}(0,\Sigma^i)$. The basic step of the Essential filter is of the form

$$\begin{bmatrix} \hat{T} \\ \hat{R} \end{bmatrix}(t+1) = \begin{bmatrix} \hat{T} \\ \hat{R} \end{bmatrix}(t) + L(t) \begin{bmatrix} \vdots \\ \tilde{\mathbf{x}^i}^T(t)\mathbf{Q}(\hat{T},\hat{R})\tilde{\mathbf{x}}^i(t-1) \\ \vdots \end{bmatrix} \tag{8.2}$$

where L has the structure of the gain of an Extended Kalman Filter (EKF) [58, 55] whose states are the motion parameters and $R \in SO(3)$. The quantities

$$\epsilon^i(t) \doteq \tilde{\mathbf{x}^i}^T(t)\mathbf{Q}(\hat{T},\hat{R})\tilde{\mathbf{x}}^i(t-1) \quad \forall i = 1:N \tag{8.3}$$

are the components of the *pseudo-innovation* vector, and measure how far each point is from the current motion interpretation $(\hat{T},\hat{R})$. The Essential filter also updates the variance of the motion estimation error through a discrete Riccati equation. Since the constraint (8.1) is *linear* in $\mathbf{Q}$, we use the (improper) notation $\mathbf{x}'^{i^T}\mathbf{Q}\mathbf{x}^i \doteq \chi^i(\mathbf{x}',\mathbf{x})\mathbf{Q} = 0$. Once N points are observed we can stack the measurements into a $N \times 9$ matrix χ and write $\chi\mathbf{Q} = 0$. We also use the shorthand $\hat{\mathbf{Q}}$ for $\mathbf{Q}(\hat{T},\hat{R})$. The matrix $\mathbf{Q}$ belongs to the the so called "Essential manifold" (see chapter 3 for details).

Consistency with the rigidity assumption

Suppose at time t the filter is in steady-state operation, and is estimating a rigid motion with some innovation norm (typically on the order of 10^{-2} to 10^{-4}). Suppose at time $t+1$ some points enter the scene which do not belong to that rigid motion. At time t the filter has produced the best prediction of motion at time $t+1$ given the measurements up to time t: $\hat{\mathbf{Q}}(t+1|t)$. We can therefore make a "prediction" of the innovation process $\hat{\epsilon}^i(t+1|t) \doteq \chi(\mathbf{x}', \mathbf{x})\hat{\mathbf{Q}}(t+1|t)$ and compare each component against the variance of the previous innovation: $\sigma_\epsilon^2(t)$.

In our implementation we reject at each time all the points which produce a residual error $\hat{\epsilon}^i(t+1|t)$ greater than one standard deviation of the innovation. Furthermore we can include into the filter any point which comes into the scene and produces a residual error within a standard deviation of the innovation. This allows dealing easily with occlusion, appearance of new feature points and splitting of rigid objects. It is also possible to perform robust statistics on the components of the innovation, although we find that the simple test proposed works well enough on the sequences we have tried.

The above discussion relies on the assumption that the filter is in steady state operation, hence estimating the motion of a single moving object. What can we do at the initial time, when we have no clue of what the motions in the scenes are? We will show in the next sections how the innovation test can be exploited to initialize a filter for each moving object.

8.2 Clustering and initialization

At the initial time instant we have a set of points and their correspondents at subsequent time instants (see chapter A). The first thing one is tempted to do is to run a filter until it converges to some "dominant" motion, rejecting progressively all the points which are not compatible, then assign the rejected points to a new filter, and so on, until all the points are assigned to a filter. However, the Essential scheme is very sensitive to the presence of outliers (which is the key of the regime mode), and it does not converge if more than few points are inconsistent with a single rigid motion interpretation. Furthermore, the innovation test can be performed *only when convergence has been reached*: otherwise, the norm of the innovation is large, which causes all the points to be rejected.

The separation mode is in charge of constructing a number of "clusters" of points which are likely to belong to the same rigid object, based only on 3-D motion (hence not exploiting local 2-D cues). The initialization phase runs a filter for each cluster and merges the clusters that have converged to similar motions.

Separation of initial motions

Let us examine the structure of the innovation (or residual) ϵ. It is the image of χ via $\hat{\mathbf{Q}}$, considered as an element of the vector space $\mathbb{R}^9$. If all the N points which build up χ were part of a rigid body, and no noise was present, then $\hat{\mathbf{Q}}$ would span the null space of χ and the residual error would be zero. Suppose a point $\mathbf{x}^i$ is added which does not belong to the rigid motion, then the corresponding component of the residual error $\epsilon^i \doteq \chi^i \hat{\mathbf{Q}}$ is greater than zero and the point can be easily spotted. However, we do not know $\hat{\mathbf{Q}}$, and in fact there might be many objects moving, each with its corresponding motion $\hat{\mathbf{Q}}$. Now suppose two objects are undergoing independent and unknown motions. The matrix χ has now full rank [79]. Let us define the *"residual space"* as the span of χ. The intuition is that, if we pick up an arbitrary motion $\tilde{\mathbf{Q}}$, the errors $\chi\tilde{\mathbf{Q}}$ in the residual space corresponding to points which belong to the same motions tend to cluster. For example when $\tilde{\mathbf{Q}}$ is very close to the motion of one of the two objects, its points will produce a very small residual, while others will have larger errors. We want to explore experimentally the possibility of using a similar criterion for separating points based on their residual errors.

One could think of computing residuals with respect to an arbitrary motion set $< \mathbf{Q}^i >_{\{i=1:K\}}$ for grouping points which are associated by similar rigid motions. A question of *sufficient excitation* arises about the family of motions one chooses [94]. If the family $< \mathbf{Q}^i >_{\{i=1:K\}}$ is chosen properly, points corresponding to different rigid motions will group into different clusters in the residual space. Which family of motion vectors do we use? how do we perform the clustering if separation occurs?

Our choice for the family $< \mathbf{Q}^i >_{\{i=1:K\}}$ is the canonical basis of the motion space $\mathbb{R}^5$ lifted to the Essential manifold [89]. This choice, although simple, may be far from the optimal. Another simple choice of sufficiently exciting motions are random vectors in $\mathbb{R}^5$ lifted to the Essential manifold. We could also employ the canonical basis (or random vectors) in $\mathbb{R}^9$, although they may represent points which are not on the Essential manifold.

Given any basis of K elements, for each measurement set χ we produce a matrix $\mathcal{E} \doteq [\epsilon_1, \epsilon_2, ...\epsilon_K]$. Then we cluster the points using a nearest neighbor criterion *in the residual space.* To do so, we produce a matrix $D = \{d_{i,j}\}$ measuring the distance of the error vectors corresponding to each couple of points: $d_{i,j} = \|\epsilon^i - \epsilon^j\|$. D is a $N \times N$ matrix, called the separation matrix. We mark for each point i (row) all the points j (columns) which have an error smaller than a threshold: $d_{i,j} \leq \gamma$. In our experiments we have used $\gamma = 3\ mean(\Delta)$, where Δ is the vector having as its elements i the minimum distance of the point i from the other points.

We have tested the separating power of this procedure on a variety of motions and points configurations. We have evaluated roughly as 0.3 the probability of having clusters which contain no spurious points and more than 40 % of the correct points. Therefore out of 100 clusters generated (one about each point), 30 contain at least 40 points which are moving with a coherent rigid motion. The Essential filters initialized for such sets converge from an arbitrary initial condition. Some instances are reported in the experimental section.

Initialization phase

The separation procedure has produced N clusters of points. For each of these clusters we start an Essential filter. According to the estimates of the separation step, for 100 clusters, one about each point, 30 will have a set of at least 40 points all belonging to the same rigid motion. We initialize each filter with one step of the basic Longuet-Higgins algorithm (see chapter 3).

After some settling time (20 steps) we evaluate the norm of the innovation process for each filter. We discard filters with high innovation norm (≥ 1), and we merge together points belonging to clusters which have produced motions whose difference is in the range of a standard deviation of the estimation error. At this point we have initialized the algorithm and we have one Essential filter associated to each rigid cluster.

8.3 A practical study

In this section we will show the results of some experiments on the operation of the segmentation scheme. We will show each mode of operation separately: first the performance of the *separation* step is tested on a synthetic set of transparent clouds of points rotating about two orthogonal axes. The same is then repeated

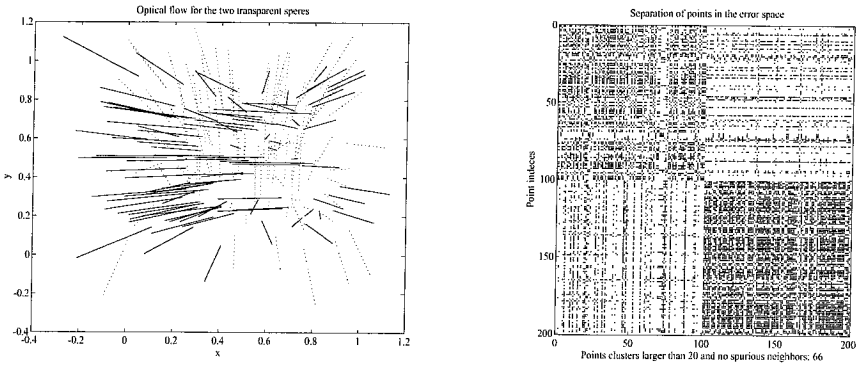

Figure 8.2: *(left) Optical flow generated by two clouds of points rotating about two orthogonal axes. Points belonging to one cloud are plotted with dotted lines, while the other cloud is plotted in solid lines. (right) Separation matrix. For each point (row) we mark a dot on each other point (column) for which the difference of the residuals ($d_{i,j}$) is smaller than a threshold. The points belonging to one object are ordered from row 1 to row 100, while points of the second object are labeled from 101 to 200. Ideally we would like to see two black diagonal blocks, meaning that each cluster contains all and only the points moving coherently. This does not happen in the experiments; however, the number of clusters having no spurious neighbors and collecting more than 20 points are 66 out of 200 (circa 30%).*

when the two clouds are rotating *about the same axis* in opposite directions (Ullmann's experiment [103, 102]). Then the *initialization mode* is tested on typical sets of points of the rotating clouds. We show the convergence of a filter associated to a cluster containing no spurious points and the divergence of a filter attached to a cluster with 20 % of spurious points. We then show the behavior of the *regime* phase when a rigid object attached to a filter splits into two objects which move with independent motions.

Throughout the experiments we have used initial information about the scale factor (norm of initial translation or distance from the centroid) and then propagated it through the estimation procedure. In the synthetic sequences the images are generated by a simulation program which adds Gaussian noise to the image plane measurements with 1 pixel std, according to the performance of the most common feature tracking and optical flow schemes [6].

8.3.1 Separation

Transparent objects rotating about orthogonal axes

Two clouds of points in the same 3D region undergo a rotational motion about two orthogonal axes. An example of an optical flow generated by this sequence is shown in fig. 8.2 (left). As it can be seen the two clusters can be separated quite easily based on the direction of the 2D flow. However, neighboring points

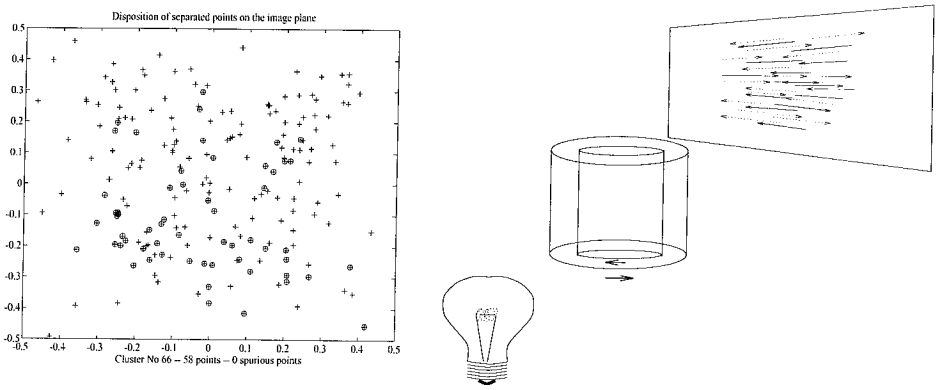

Figure 8.3: *(left) Distribution of selected points (circled) on the image plane. It can be seen that the selected points are mixed with points which belong to the other motion. (right) Illustration of the Ullmann experiment. Two transparent cylinders rotate about the same axis and in opposite directions. The only cue for segmentation is three dimensional motion.*

moving with the same 3D motion can have opposite 2D velocity. In fig. 8.2 (right) is shown the matrix D described in section 8.2 (the separation matrix). Points satisfying the neighboring criterion in the residual space are marked as dots. In this example points from 1 to 100 belong to one object, and from 101 to 200 belong to the object rotating about the orthogonal axis. Hence in an ideal situation we expect a symmetric, block diagonal structure with zeros on the off-diagonal blocks. Instead, the number of clusters having no spurious neighbors and collecting more than 20 points are 66 out of 200 (circa 30%). Hence for 200 filters which run independently in the initialization phase, 66 will converge to a rigid motion. In fig. 8.3 (left) we show an image plane view of the selected points for the one of the clusters. It can be seen that the selected points are mixed with other points which belong to the orthogonal motion.

Transparent objects rotating about the same axis with different directions

The same experiment described in the previous section is repeated when the two clouds of points are rotating about the same axis in opposite directions (see figure 8.3 right). Psychophysical experiments showed that this is a difficult task for humans; 3D motion is the only available cue.

The image plane view is reported in fig. 8.4 (left), and the corresponding separation matrix D in fig. 8.4 (right). The number of clusters collecting no spurious neighbors is smaller than in the previous experiment. However, the number of pure clusters with more than 20 points is still 12, which corresponds to 5% circa of the original feature set. Filters initiated with one of the 12 pure (rigid) clusters converge to the proper

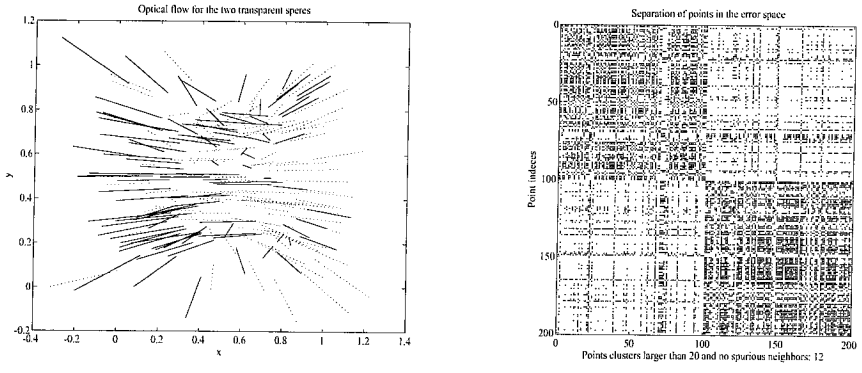

Figure 8.4: *(left) Optical flow generated by the Ullmann experiments. Two clouds are rotated about the same axis in opposite directions. Observe that in this case no region-based algorithm could work and 3D "transparent" motion is the only available cue. (right) Separation matrix. The number of pure clusters with more than 20 points is 12, which corresponds to 5% circa of the original feature set.*

motion allowing the scheme to be initialized correctly.

8.3.2 Initialization

In this section we show a prototype of a converging cluster (fig. 8.5 left) and a diverging one (fig. 8.5 right). Motion is represented using six components (three of translation and three of rotational velocity); ground truth is shown in dotted lined.

8.3.3 Regime: a motion splitting experiment

In this section we show an experiment of a splitting object: one of the clouds of points is rotating and a regime filter is tracking its motion. After 25 frames the cloud breaks into two sets of points: one keeps on rotating with the same motion, while the other starts rotating about an orthogonal axis. All the points which belong to the split cloud are rejected by the filter. Since all of them belong to the same rigid motion, the new filter initialized with the rejected points converges rapidly to the motion of the new split cloud. In fig. 8.6 (left) we show the motion for the cluster which continues after the splitting, and in fig. 8.6 (right) we show the motion estimates for the split cloud.

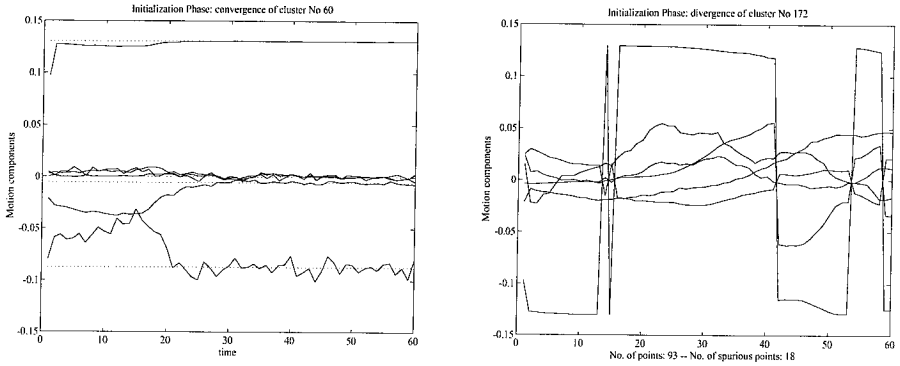

Figure 8.5: *Initialization phase: convergence (left) or divergence (right) of clusters of points. The motion coordinates (three for rotation and three for translation) are plotted in solid lines as estimated in the initialization phase. The behavior of a typical converging cluster and a typical diverging one is plotted. Ground truth is in dotted lines. Note that 20 steps are sufficient for deciding whether a filter has converged or not. Also note that the diverging cluster has 18 spurious points out or 93, i.e. circa 20%, which is sufficient not to reach convergence on the "dominant motion".*

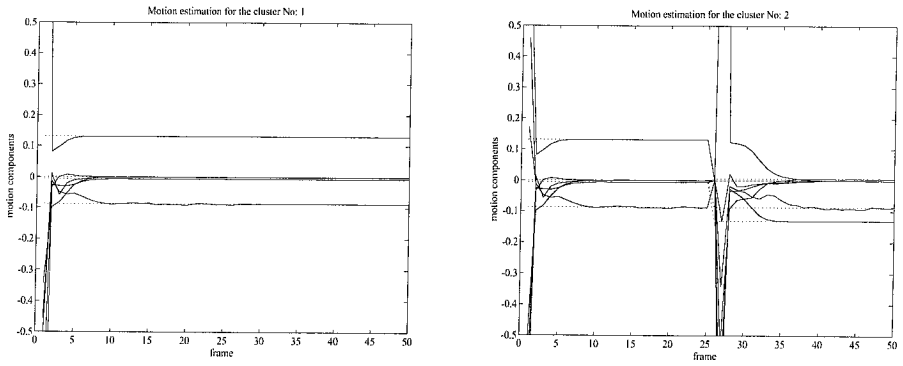

Figure 8.6: *Motion estimates for the splitting experiment: cluster of points with continuous motion (left) and split cluster (right). Filter estimates (solid) vs. ground truth (dotted).*

Chapter 9 Dynamic calibration

Motion estimation from image sequences is usually performed in two steps: first the camera is calibrated, in order to establish metric relationships between world coordinates and image-plane measurements. The internal parameters (pixel size, optical center, focal length), are usually estimated *off-line*. Once calibration is performed, one can estimate camera motion and ambient structure in a variety of ways, as we have described in previous chapters.

Most of the recursive motion estimation schemes rely upon the exact knowledge of internal camera parameters. However, experimental evidence shows that these can change drastically during a long sequence [22] due to zooming and changing of the aperture. Moreover, *often it is not possible to access the physical device* which produced the sequence.

Many approaches for camera calibration are available in the literature; they can roughly be classified as:

1. Batch schemes, which rely on the knowledge of the *structure* by including a calibration rig in the field of view (see [67] and references therein).

2. Active devices, which rely on the knowledge of the camera *motion* by controlling the configuration (pose) of the camera [29, 22, 7].

3. *Arbitrary structure and motion.* Camera self calibration is performed along with motion estimation [32].

The first two approaches assume that the camera is available for measurements, by either controlling its motion or inserting a known object into the field of view. Therefore, it seems that the third approach is the only feasible solution when the the device which produced the sequence is not available, as for example in image compression applications or automation of image processing tasks for the movie industry.

Faugeras et al. [32] propose a *batch* scheme which reconstructs the epipolar transformation of the camera, and then imposes the structure of such a transformation by solving a set of polynomial equations, known as Kruppa's equations. However, the scheme has some drawbacks which make it unattractive for real world applications. In particular

- High sensitivity to pixel-noise
- Numerical instability
- Motion parameters and internal parameters are treated alike. While camera-motion can vary arbitrarily during a sequence, it is conceivable that some parameters (for example the pixel size or aspect ratio) are constant over long periods of time
- Not all the information coming from a sequence is exploited. The scheme processes 3 images at a time and does not use temporal coherence or a-priori information (such as reference values for focal length, initial confidence in the position of the optical center etc.).

Hence we want a recursive scheme which, after each incoming image, updates the computation performed at the previous step. We also want the scheme to be *causal* so that it can be used for real-time implementations.

Background and notation

In this chapter we will adopt the same notation of the "Essential filter" introduced in chapter 3. Note that the same considerations also apply for the other models described in previous chapters: the state-space can be extended and the calibration parameters estimated on-line, provided that the corresponding model is observable.

Outline of the chapter

In this chapter we present a scheme for performing ego-motion estimation and camera calibration recursively and causally for an image sequence. It does not need a calibration rig nor to control motion, while it exploits redundancy at each step and computations from each previous step by recursion. A priori information about calibration can be used, if available, as initial conditions for the estimation scheme. Internal parameter time constants are adjustable by tuning their random walk models.

The scheme is based upon a modification of the "Essential filter", extended to estimate camera parameters according to the representation of [32]. A key feature is that the structure of the epipolar geometry is imposed explicitly as the structure of the state-space of the filter, so we do not need to solve explicitly complicated polynomial equations in order to enforce such a structure. From a different point of view, our filter can be viewed as a recursive differential scheme for solving Kruppa's equations.

9.1 Camera model: internal and external parameters

The camera may be modeled as a perspective projection map

$$\begin{aligned} M : \mathbb{R}^3 &\rightarrow \mathbb{R}^2 \\ \mathbf{X} &\mapsto \mathbf{x}. \end{aligned} \tag{9.1}$$

The simplest instance, which we have used so far, is the so called "ideal pinhole model":

$$\mathbf{X} \doteq \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^T \mapsto \begin{bmatrix} x \\ y \end{bmatrix}^T \doteq \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{bmatrix}^T \doteq \mathbf{x}. \tag{9.2}$$

It can also be represented as a *linear map* between real projective spaces, $\bar{M} : \mathbb{RP}^3 \to \mathbb{RP}^2$: in homogeneous coordinates it is represented by a 3×4 matrix $\begin{bmatrix} \mathbf{A} & | & 0 \end{bmatrix}$ where

$$\mathbf{A} \doteq \begin{bmatrix} fs_x & 0 & -i_0 \\ 0 & fs_y & -j_0 \\ 0 & 0 & 1 \end{bmatrix}$$

is the internal parameter matrix. f is the focal length, (i_0, j_0) the coordinates of the optical center and (s_x, s_y) the pixel sizes along the image plane coordinates. The deviation from 90^o of the angle between the optical axis and the CCD surface is usually on the order of 1^0, and we may therefore neglect it.

In the case of an uncalibrated camera, a constraint similar to Longuet-Higgins' coplanarity constraint can be derived simply as follows: given the projection of a feature $\mathbf{x}(t)$ at time t, its correspondent at $t+1$, $\mathbf{x}(t+1)$, must lie on the epipolar line ${}^t\mathbf{e}_{t+1}$ [1]. Such a line is described in projective coordinates by a linear function of $\mathbf{x}(t)$. The matrix representing such a linear function is called the *fundamental matrix* $\mathbf{F}$, which is *defined* by the relation ${}^t\mathbf{e}_{t+1} \doteq \mathbf{F}\mathbf{x}(t)$. It can be easily verified that

$$\mathbf{F} \doteq \mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}, \tag{9.3}$$

where $\mathbf{Q}$ is an Essential matrix. From the definition of the epipolar line, one may derive a generalization of the Essential constraint seen in chapter 3 as

$$\mathbf{x'}^{iT}\mathbf{F}\mathbf{x}^i = 0 \qquad \forall i = 1 \ldots N. \tag{9.4}$$

[1] The epipolar line corresponding to a given feature is the intersection of the epipolar plane of that feature and the image-plane. The epipolar plane is the plane determined by the two centers of projection and the feature in question.

The scheme presented in [32] consists in first estimating $\mathbf{F}$ from (9.3), and then imposing its structure a-posteriori by solving the Kruppa equations, which correspond to enforcing the fact that $\mathbf{A}^T\mathbf{F}\mathbf{A}$ (is Essential and therefore) has two equal singular values and zero determinant.

9.2 Essential filters for fundamental matrices

The Essential filter has been introduced in order to identify the dynamical model determined by the Essential constraint:

$$\begin{cases} \mathbf{x}^{i^T}(t+1)\mathbf{Q}(t)\mathbf{x}^i(t) = 0 \\ \tilde{\mathbf{x}}^i(t) = \mathbf{x}^i(t) + n^i(t) \end{cases} \qquad \forall i = 1 \dots N. \tag{9.5}$$

We propose to extend the Essential filter to estimate fundamental matrices, by just substituting $\mathbf{Q}$ with $\mathbf{F}$, and *impose the structure of the fundamental matrix explicitly* by writing the estimator in local coordinates: the estimate at each step determines a matrix which is *fundamental by construction*, and we do not need to enforce the structure by solving explicitly ill-conditioned polynomial equations. The structure of resulting update is very similar to the Essential filter [89]:

$$\begin{bmatrix} \hat{\xi} \\ \hat{T} \\ \hat{R} \end{bmatrix}(t+1) = \begin{bmatrix} \hat{\xi} \\ \hat{T} \\ \hat{R} \end{bmatrix}(t) + L(t) \begin{bmatrix} \vdots \\ \tilde{\mathbf{x}^i}^T(t)\mathbf{A}^{-T}(\hat{\xi})\mathbf{Q}(\hat{T},\hat{R})\mathbf{A}^{-1}(\hat{\xi})\tilde{\mathbf{x}}^i(t-1) \\ \vdots \end{bmatrix} \tag{9.6}$$

where $\xi \doteq [fs_x, fs_y, i_0, j_0]^T$; L has the structure of the gain of an Implicit Extended Kalman Filter (IEKF) (see appendix F).

If we call $\alpha \doteq [\xi\ \theta\ \phi\ \Omega]^T \in \mathbb{R}^9$, where Ω are the exponential coordinates of $R = e^{\Omega\wedge}$, and (θ, ϕ) are the spherical coordinates of T, then we can write the complete set of equations for the filter:

Prediction step

$$\begin{cases} \hat{\alpha}(t+1|t) = \hat{\alpha}(t|t) & \hat{\alpha}(0|0) = \alpha_0 \\ P(t+1|t) = P(t|t) + \Sigma_\alpha(t) & P(0|0) = P_0 \end{cases}$$

Update step

$$\begin{cases} \hat{\alpha}(t+1|t+1) = \hat{\alpha}(t+1|t) + \\ \quad L(t+1)\tilde{\mathbf{x}^i}^T(t)\mathbf{A}^{-T}\mathbf{Q}(\hat{\alpha}(t+1|t))\mathbf{A}^{-1}\tilde{\mathbf{x}}^i(t-1) \\ P(t+1|t+1) = \\ \quad \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + \\ \quad L(t+1)D_+(t)\Sigma_n(t+1)D_+^T(t)L^T(t+1) \end{cases}$$

where

$$\begin{cases} L(t+1) = P(t+1|t)C^T(t+1)\Lambda^{-1}(t+1) \\ \Lambda(t+1) = C(t+1)P(t+1|t)C^T(t+1) + \\ \qquad +D_+(t+1)\Sigma_n(t+1)D_+^T(t+1) \\ \Gamma(t+1) = I - L(t+1)C(t+1) \\ D_+(t+1) \doteq \left(\frac{\partial \mathbf{x}^{i\,T}(t+1)\mathbf{Q}(t)\mathbf{x}^i(t)}{\partial \mathbf{x}(t+1)}\right)_{|\tilde{\mathbf{x}}(t),\hat{\alpha}(t)} \\ C(t+1) \doteq \left(\frac{\partial \mathbf{x}^{i\,T}(t+1)\mathbf{Q}(t)\mathbf{x}^i(t)}{\partial \alpha(t)}\right)_{|\tilde{\mathbf{x}}(t),\hat{\alpha}(t)} \end{cases}$$

where Σ_α and Σ_n denote the variance of the noises $\alpha(t)$ and $n(t)$ respectively.

9.3 Tradeoffs and sufficient excitation

In order for the motion and calibration parameters to be estimated, one must verify that the model just described is observable. As it turns out, there are some tradeoffs due to the coupling between calibration and motion parameters. For instance, the localization of the optical center is strongly coupled with the direction of translation (with its component on the image plane).

We have performed a set of simulations on a noisy synthetic sequence, which show that often the estimates are subject to biases when the structure and the motion are not "sufficiently exciting". In figure 9.1 we show the estimates of the translation and rotation parameters. In figure 9.2 we show the estimates of the internal parameters. The noise on the image-plane was one tenth of a pixel, according to the performance of the best optical flow/feature tracking techniques [6]. Convergence is reached in about 100 frames. Each iteration consists of about 100 Kflops: an implementation using Matlab (not optimized) runs at $.6Hz$ on a Sparc 10-20.

Note that, once the motion has been reconstructed, we may feed the estimates onto any Structure-

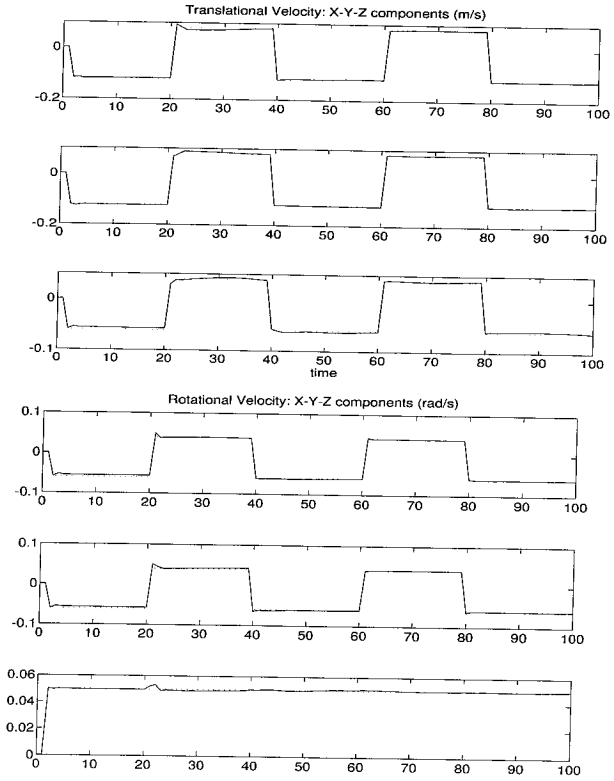

Figure 9.1: *(Top) Translational velocity: filter estimates (solid) vs. true values (dotted). (Bottom) Components of rotational velocity.*

From-Motion module that processes motion error [84, 93]. However, the motion configurations that allow estimating accurately the scene structure, as for example fronto-parallel translation, are often not sufficiently exciting for estimating the camera parameters. Vice-versa, motions that allow a good estimation of the camera calibration are often ill-conditioned for estimating depth, as for example a spiral along the optical axis. Therefore, there is also an intrinsic conflict between the estimation of the camera parameters and the structure of the scene.

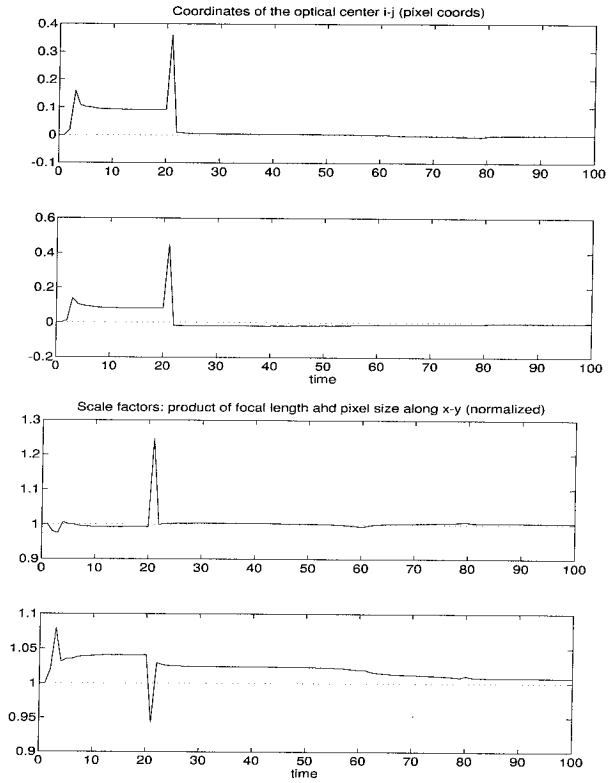

Figure 9.2: *(Top) Coordinates of the center of projection: filter estimates vs. true values. (Bottom) Pixel size along image coordinates*

Chapter 10 Visual motion control

Traditionally, the control task in systems using vision as a sensor has been formulated directly on the image-plane [42]. This choice is very natural in certain applications, for example tracking, docking, navigation etc.. However, it results in methods that are intrinsically local, whereas there are applications in which one is required to track a globally prescribed path in the full configuration space. Furthermore, the control on the image-plane exhibits some limitations due to the dependence of the controller on the structure (depth) of the observed scene.

Background and notation

See chapters 3 and 5.

Outline of the chapter

In the next section we briefly describe a simple tracking control on the image-plane, and highlight its limitations. In the following section we propose to formulate the tracking problem in the configuration space in its Essential representation (through the Essential manifold). We anticipate that the resulting control has more "global" features and does not depend on the structure of the observed scene. Instead, structure comes as a byproduct of the Essential estimator once the control task has been accomplished. In the experimental section we describe simulated experiments of the behavior of a simple controller based on the Essential filter.

We suppose the camera is mounted on a moving platform, on which we have full control. For simplicity we neglect the dynamic constraints and assume to be able to control directly the translational and rotational velocity of the platform. Suppose our task is to maintain a given relative configuration between the platform and the scene. Such a situation occurs in tracking the motion of a three-dimensional object (of unknown shape and kinematics) or in maintaining a fixed pose with respect to a scene despite the action of disturbances on the platform (as for example in hovering or in underwater operation).

10.1 Control on the image-plane

Consider any of the models described in chapter 2. The time derivative of their output $\mathbf{x}$ (also called "motion field" and approximated by the optical flow) can be written as:

$$\dot{\mathbf{x}}^i(t) = \mathcal{J}(\mathbf{x}^i(t), \mathbf{X}_3^i(t))u(t) \quad \forall i = 1 \dots N \tag{10.1}$$

where

$$\mathcal{J}(\mathbf{x}, \mathbf{X}_3) \doteq \begin{bmatrix} \frac{1}{X_3} & 0 & -\frac{x_1}{X_3} - x_1 x_2 & 1 + x_1^2 & -x_2 \\ 0 & \frac{1}{X_3} & -\frac{x_2}{X_3} - (1 + x_2^2) & x_1 x_2 & x_1 \end{bmatrix} \tag{10.2}$$

and $\mathbf{x}^i$ indicates the image-plane coordinates of the projection, while $\mathbf{X}_3^i$ denotes the third component of the space coordinate (depth) of each point. The vector $u(t) \doteq (V(t), \Omega(t))$ is the canonical exponential representation of the instantaneous motion $(T(t), R(t))$. Suppose the initial configuration of the points on the image-plane is $\mathbf{x}^i(t_0|t_0) = \mathbf{x}_0^i$, and an exogenous agent acts by moving either the platform on which the camera is mounted or the target which the camera is looking at, producing a deformation of its image:

$$\mathbf{x}^i(t+1) = \mathbf{x}^i(t) + \tilde{\mathbf{x}}^i(t). \tag{10.3}$$

Suppose our goal is to keep the configuration of the observed points fixed at the value of the initial instant $\mathbf{x}_0^i$. At any time we can measure a noisy version of the instantaneous configuration modified by the external agent, and act with the control of the platform on which the camera is mounted. Using a first-step approximation, one could write

$$\mathbf{x}^i(t+1) \cong \mathbf{x}^i(t) + \mathcal{J}(\mathbf{x}^i(t), \hat{\mathbf{X}}_3^i(t))u(t) \tag{10.4}$$

and use a one-step deadbeat controller:

$$u(t) \doteq \mathcal{J}^\dagger(\mathbf{x}^i(t), \hat{\mathbf{X}}_3^i(t)) \cdot (\mathbf{x}_0^i - \mathbf{x}^i(t)) \tag{10.5}$$

where $\dagger$ denotes the pseudoinverse. Note that the control depends on the depth of each point of the scene $\hat{\mathbf{X}}_3^i(t)$. Such a strategy has been experimented by Kimura et al. [42], who pioneered the control on the image-plane. However, the expression of the deadbeat controller on the image-plane depends on the inverse depth of each visible points, which needs to be "estimated" on line. This problem can be overcome by assuming that *the structure of the scene is known*, and therefore the inverse depth can be recovered linearly (via calibration as in Appendix A). Another alternative, which we do not pursue here, is the use of a stereo system.

If the structure (depth) of the scene is not known, we need to *estimate* it, unless the motion of the target

is purely rotational about the center of the viewer's reference, in which case $\mathcal{J}$ does not depend on the depth. In order to estimate depth, we need non-zero *disparity* (also called visual parallax), which is the displacement of corresponding points across different images. When disparity is close to zero, the recovery of the depth is ill-conditioned (see Chapter 4). Therefore the image-based controller, which depends on the depth, tries to drive the system towards a configuration of zero disparity, which does not allow recovery of depth. As a result the controller either "drifts" or "swings", as is discussed in the experimental section.

10.2 Control on the Essential manifold

Consider $\mathbf{Q}_0 \in E$ describing the relative configuration between the scene and the platform at the initial instance, and suppose we ask it to be constant despite the motion of the scene, encoded by an arbitrary $d(t) \in E$. We indicate with $\mathbf{Q}(t)$ the Essential matrix describing the motion between the *initial* instant and the current time, which is therefore defined by the Essential constraint $\mathbf{x}^i(t)^T\mathbf{Q}(t)\mathbf{x}_0^i \doteq 0$. The effect of the exogenous displacement (motion of the scene) and the control action are described by the model

$$\begin{cases} \mathbf{Q}(t+1) = \mathbf{Q}(t) \oplus \Phi^{-1}(u(t)) \oplus d(t) \\ y^i(t)^T\mathbf{Q}(t)y(0) = \tilde{n}(t) \end{cases} \tag{10.6}$$

where $\oplus$ represents the sum of the local coordinates, $\tilde{n}$ describes the effect of the estimation error (it is in fact the pseudo-innovation of the Essential filter). In general we may want to specify the control task in terms of some *distance* defined on the Essential space, $d_E(\mathbf{Q}_1, \mathbf{Q}_2)$, so that

$$e(t) \doteq d_E(\mathbf{Q}(t), \mathbf{Q}_d(t)) \tag{10.7}$$

satisfies a difference equation whose dynamics can be assigned by choice of the input.

10.2.1 Choice of a metric on the Essential manifold

Since E can be interpreted as an alternative representation of $SE(3)$, any control strategy on the Euclidean group can be mapped onto the Essential manifold. However, if we were able to formulate the control strategy directly on the Essential manifold, the Essential filter would then gives us a direct estimate of the full state

which is optimal, independent of the structure and obtained linearly from the visual data [89].

The choice of a metric on the Essential space is not a trivial issue, and we intend in this section to hint at some possible choices. First of all any metric in the Euclidean space $SE(3)$ can be "mapped" onto the Essential manifold by defining

$$d_E(\mathbf{Q}_1, \mathbf{Q}_2) \doteq d_{SE(3)}(\Psi^{-1} \circ \Phi(\mathbf{Q}_1), \Psi^{-1} \circ \Phi(\mathbf{Q}_2)) \tag{10.8}$$

where Ψ and Φ are local coordinatizations of $SE(3)$ and E respectively. An alternative (and equivalent) method is to set the metric directly in the local coordinates and then "lift" it to the manifold. It must be pointed out, however, that there is no natural (bi-invariant) choice of a metric on the Euclidean group. Another possibility is to "project" a metric of the ambient space of the Essential manifold, $\mathbb{R}^9$, by using the projection onto the manifold pr_E. It is unclear at the moment what the properties of such a metric may be. Note also that a possible way of generating a path between two points of the Essential manifold, based on its interpretation as the tangent bundle of $SO(3)$, is to formulate a control that connects two points of $SO(3)$ with a given direction in the tangent plane. Such control strategies, called "dynamic interpolation" have been studied for Riemannian manifolds and Lie groups in [21, 47].

10.2.2 Minimum-time, structure independent control on the Essential manifold

In this section we consider a simple experiment: we want to formulate the control that drives the relative configuration to the desired one in one step (the minimum time in this discrete-time framework), as we have done in section 10.1 for the control on the image-plane. We do not make any assumption on the scene, and we want to develop a control strategy which is independent on depth, so that we do not have ill-conditioned controllers at unobservable configurations of the system.

The model described in eq. (4.38) gives an immediate expression for such a minimum-time controller. Suppose we are only interested in maintaining the initial configuration, then $\mathbf{Q}(t_0) = 0$, and our control can be inferred from the current estimate of the essential matrix, $\hat{\mathbf{Q}}(t+1|t)$, by

$$\mathbf{Q}(t+1|t) = \mathbf{Q}(t|t) \oplus d(t) \tag{10.9}$$

$$\mathbf{x}^i(t+1)^T(\hat{\mathbf{Q}}(t+1|t) + n(t))\mathbf{x}_0^i = 0 \tag{10.10}$$

$$\Phi^{-1}(u(t+1)) \doteq -\hat{\mathbf{Q}}(t+1|t) \tag{10.11}$$

$$\mathbf{Q}(t+1|t+1) = \mathbf{Q}(t+1|t) \oplus \Phi^{-1}(u(t+1)) = n(t) \tag{10.12}$$

and therefore, provided that our estimator is unbiased, the control

$$u(t) = -\Phi(\hat{\mathbf{Q}}(t|t-1)) \tag{10.13}$$

gives a one-step correction which brings the state to the goal instantaneously up to white, zero-mean noise.

10.3 Some practical experiments

In this section we report an experiment of simple control laws for maintaining a given relative configuration between a scene and an actuated platform on which the camera is mounted.

In figure 10.1, we have simulated a rigid cloud of points moving in front of the camera, which is mounted on some actuated platform, and have generated a simple minimum-time control, based on the motion estimated by the Essential filter, in order to maintain the initial configuration between the camera and the scene. The following experiment, reported in figure 10.2, describes a similar experiment for a different motion of the scene.

In the last experiment, reported in figure 10.3, we have implemented a minimum-time image-plane control designed for the same task of the previous experiment. In this case the controller is asked to maintain the initial configuration of the points observed on the image-plane. Therefore, at each step the controller drives the disparity (difference between projections of the same point at subsequent times, also called visual parallax) to zero. However, we have seen that the image-plane minimum-time controller *depends on the depth* (structure) of each point of the scene. When the *structure is known*, then the controller performs similarly to the one on the Essential space (see figure 10.3 (A)-(B)). If the geometry of the scene is not known, then it must be estimated. However, depth cannot be estimated for zero parallax. Therefore, the system is affected by the intrinsic conflict between *trying to drive the parallax to zero, and at the same time trying to keep it large enough* in order to be able to compute depth. The effect, which is visible in figure 10.3 (C)-(D), is that the controller "drifts" in order to accumulate a residual which is large enough for computing depth.

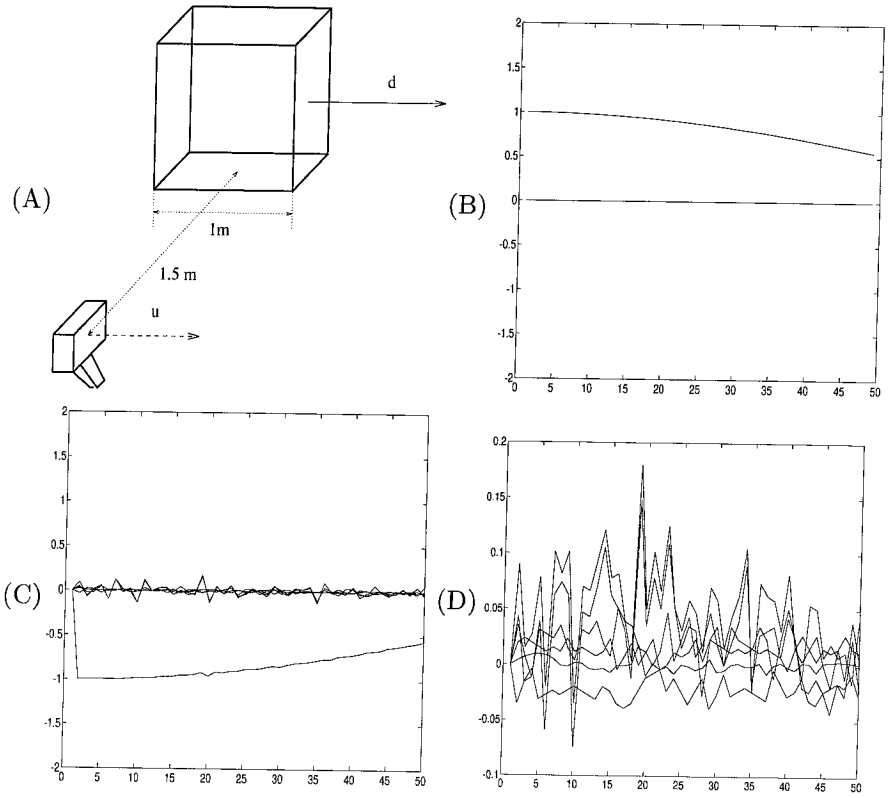

Figure 10.1: *"Configuration tracking experiment on the Essential space:* **pure translation***": (A) a synthetic scene composed of 30 feature points translates with decreasing translational velocity, the components of which are plotted in (B) in m/s. The minimum-time control, whose components are plotted in (C) in m/s, is obtained by feedback from the instantaneous estimate of the relative configuration between the scene and the camera, and quantized at 8 bits. The noise in the image-plane was additive white Gaussian with standard deviation corresponding to 10 pixels. The actuators are controlled as to maintain the initial relative configuration between the viewer and the scene; the six local coordinates of the error from the desired configuration are plotted in figure (D) (units are m/s for the error in translational velocity and rad/s for the error in rotational velocity).*

In some cases the controller "swings": when the residual is large, there is enough parallax for computing the controller accurately and drive the residual to zero; at this point the controller is computed with large errors, and the residual grows again.

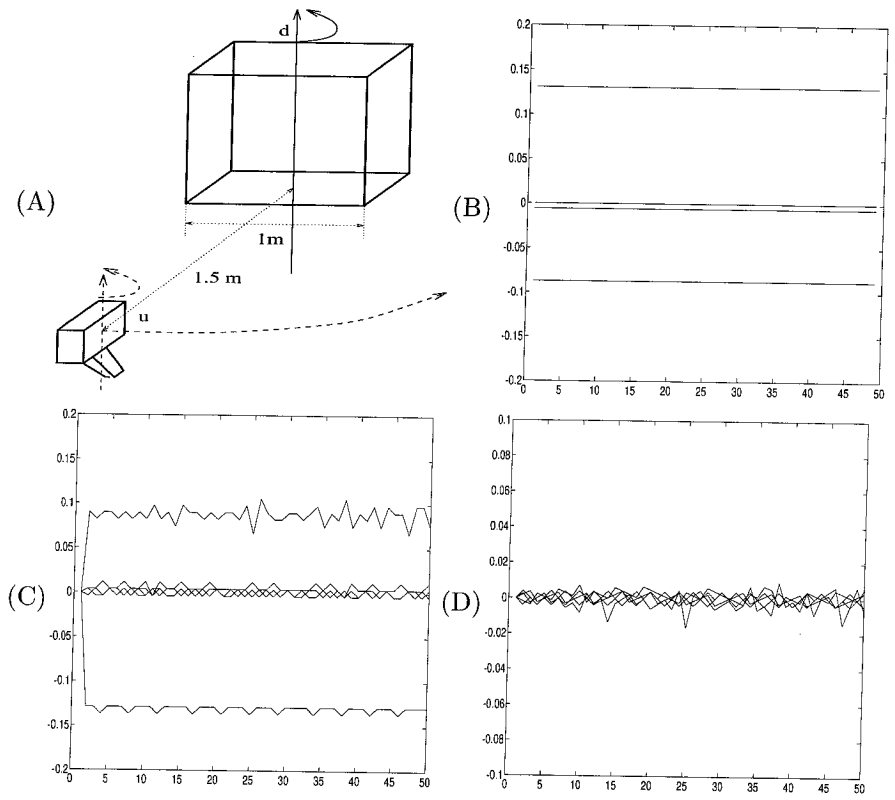

Figure 10.2: *"Configuration tracking on the Essential space:* **roto-translational motion***" (A) the scene rotates about a fixed axis which is 1.5m ahead of the observer with constant angular velocity of 5 deg/s. The local coordinates of the relative motion between the scene and the viewer in the viewer's reference are plotted in (B) (m/s for the translational velocity, rad/s for the rotational velocity). The components of the minimum-time control are plotted in (C) with the same units, and the corresponding deviation from the desired configuration is plotted in (D). The noise was white, zero-mean and Gaussian with 5 pixel std, and the controller was quantized at 8 bits.*

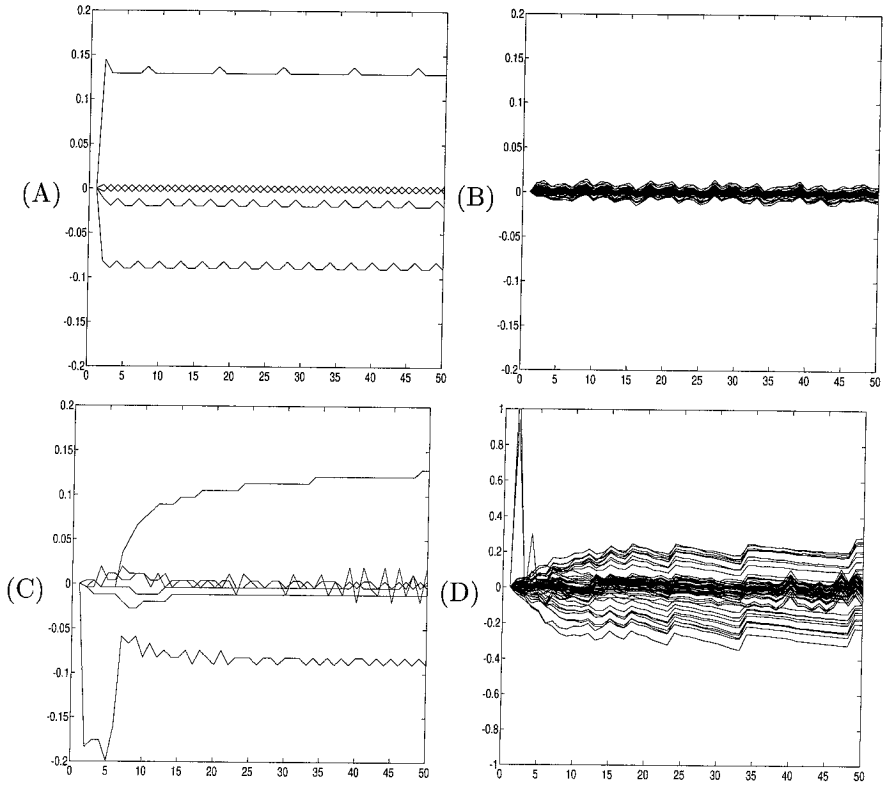

Figure 10.3: *"Configuration tracking* **on the image plane***": (A)-(B) for the same experiment described in figure 10.2, the control on the image plane* when the structure of the scene is known *(in terms of depth of each point) is comparable with the one obtained with the control on the Essential manifold, which* does not need information about the structure of the scene *(compare with figure 10.2 (C)-(D)). When the* structure of the scene is not known, *and depth has to be estimated, the control is far less robust, for it tries to drive the system to a zero-disparity configuration which is ill-conditioned (C)-(D). The controller, whose state depends on the dept¡h of the points in the scene, tries to reduce the image parallax (disparity, or residual) to zero: such configuration, however, does not allow estimating depth. The effect, which is visible in figures (C)-(D), is that the controller "drifts" in order to accumulate a residual which is large enough for computing depth.*

Part II

Implementation and Experimental Results

Chapter 11 A comparative experiment

In this chapter we compare the schemes derived in previous chapters on a common set of experiments. In order to render the discussion self-contained and avoid reference to all other chapters, we first review in synthesis the different schemes presented in chapters 2, 3, 5, 6 and 7. Then we compare the performance of all schemes on a common set of experiments on synthetic images.

11.1 Introduction

In this section we summarize the main results described in part I of the thesis in order to introduce the set of experiments described in section 11.2.

11.1.1 Modeling "Structure From Motion"

In the preceding chapters we have seen how different models for estimating motion from sequences images can be cast within the framework of dynamical systems estimation and identification. We have started from the model that is "defined" by the rigidity constraint and the perspective projection, either in a continuous-time or in a discrete-time fashion:

$$\begin{cases} \mathbf{X}^i(t+1) = R(t)\mathbf{X}^i(t) + T(t) \\ \mathbf{y}^i(t) = \pi(\mathbf{X}^i(t)) + n^i(t) \end{cases} \qquad \begin{cases} \dot{\mathbf{X}}^i = \Omega \wedge \mathbf{X}^i + V \\ \mathbf{y}^i = \pi(\mathbf{X}^i) + n^i \end{cases} \qquad \forall\, i = 1 \ldots N \tag{11.1}$$

where the *states* $\mathbf{X}^i = [X^i\ Y^i\ Z^i]^T \in \mathbb{R}^3$ are the 3-D coordinates of each of the N feature-points in the scene relative to the viewer's moving frame, $\mathbf{x} \doteq \pi(\mathbf{X}) \doteq [\frac{X}{Z}\ \frac{Y}{Z}\ 1]^T \in \mathbb{RP}^2$ represents an ideal perspective projection (pinhole), and $n^i \in \mathcal{N}(0, R^i)$ is a white, zero-mean and Gaussian measurement noise. The 3×3 rotation matrix $R(t)$ describes the change of coordinates of the viewer's moving frame between time $t+1$ and time t, and is orthonormal with positive determinant. When the rotational velocity Ω is held constant

between time samples, R is related to Ω via the exponential map: [1] $R = e^{\Omega\wedge}$. Therefore, a rotation matrix has only 3 degrees of freedom, encoded in the three-dimensional rotation vector Ω. T is a 3-dimensional vector that describes the translation of the origin of the moving frame.

It is possible to integrate the above models from the initial time-instant, and end up with an "integral" model of the form

$$\mathbf{X}^i(t) = {}^tR_{t_0}\mathbf{X}^i(t_0) + {}^tT_{t_0} \qquad \mathbf{X}^i(t_0) = \mathbf{X}_0^i \tag{11.2}$$

where the coordinates of each point relative to the initial time-frame $\mathbf{X}_0^i$ are constant, and the current configuration is described by the unknown translation ${}^tT_{t_0}$ and rotation ${}^tR_{t_0}$, relative to the initial time instant.

We have then *dynamically extended* the models above in order to include all unknown parameters T, R or V, Ω in the state-space. In order to do so, one needs to know how such parameters evolve in time. In the absence of any dynamical model, one may assume that they evolve statistically according to a random walk of some order[2]. In the case of a discrete-time first-order random walk, one ends up with the extended model

$$\begin{cases} \mathbf{X}^i(t+1) = R(t)\mathbf{X}^i(t) + T(t) \\ T(t+1) = T(t) + n_T(t) \\ R(t+1) = R(t)e^{n_R\wedge(t)} \\ \mathbf{y}^i(t) = \pi(\mathbf{X}^i(t)) + n^i(t) \end{cases} \qquad \forall\, i \;=\; 1 \ldots N(t) \tag{11.3}$$

where $n_T \in \mathbb{R}^3, n_R \in \mathbb{R}^3$ are well as $n^i \in \mathbb{R}\mathrm{P}^2$ are white, zero-mean Gaussian processes. The above model is the discrete-time version of the *structure-velocity* model introduced in chapter 2. We have applied the idea of the "reduced-order observer" [57] in order to reduce the dimension of the state by the number of the measurements, and be left with one state for each visible point, which encodes its depth in the moving

[1] The notation $\Omega\wedge$ stands for the operator that performs the vector product on $\mathbb{R}^3$: $(\Omega\wedge)\mathbf{X} \doteq \Omega \wedge \mathbf{X}\ \forall\ \mathbf{X} \in \mathbb{R}^3$. In coordinates $\Omega\wedge \doteq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$. Alternative (local) representations for rotation matrices include various types of "Euler angles"; global (embedded) representations can be obtained through unit quaternions [81].

[2] The choice of a random walk is made for shear engineering convenience, for it results in a model which is suitable for "recipe-design" of an Extended Kalman Filter.

frame. Depending on whether we use a first-order model or an integral (second-order) model, we have

$$\begin{cases} Z^i(t+1) = R_{3.}(t)Z^i(t) + T_3(t) \\ T(t+1) = T(t) + n_T(t) \\ R(t+1) = R(t)e^{n_R \wedge (t)} \\ \mathbf{y}^i(t) = \pi(Z^i(t)\mathbf{y}^i(t)) + n^i(t) \end{cases} \quad \text{or} \quad \begin{cases} Z_0^i(t+1) = Z_0^i(t) \\ \Omega(t+1) = \Omega(t) + n_\Omega(t) \\ V(t+1) = V(t) + n_V(t) \\ R(t+1) = e^{\Omega(t)\wedge} R(t) \\ T(t+1) = e^{\Omega(t)\wedge} T(t) + V(t) \\ \mathbf{y}^i(t) = \pi\left(R(t)\mathbf{y}^i(t_0)Z_0^i(t) + T(t)\right) + n^i(t) \end{cases} \tag{11.4}$$

where Z_0^i is the depth of each point at the initial time 0, which is obviously constant. Then we have pushed the idea of the reduced-order observer in order to decouple structure from the motion parameters (chapters 3 and 5), and we have applied *"output stabilization"* in order to further decouple rotation from translation (chapter 7). In all instances we have ended up with implicit dynamical models in the form

$$\begin{cases} h(\mathbf{x}^i, \alpha)\dot{\mathbf{x}}^i = 0 \\ \mathbf{y}^i = \mathbf{x}^i + n^i \quad \forall\, i = 1 \ldots N \end{cases} \qquad \alpha \in M \tag{11.5}$$

where α are unknown parameters constrained to belong to the set M. In the discrete-time case we end up with a similar form where $\mathbf{x}(t+1)$ replaces $\dot{\mathbf{x}}$. By simply changing the set M we may obtain all the different reduced models. Some relevant instances are:

Essential model: it is the well-known coplanarity constraint introduced by Longuet-Higgins [73], interpreted as a discrete-time implicit dynamical system. The unknown motion parameters T and R are encoded into a 3×3 "Essential matrix" $\mathbf{Q}$, which belongs to the space of matrices of the form $(T\wedge)R$. Such a space E is called "Essential manifold". The function h is simply $h(\mathbf{x}, \mathbf{Q}) \doteq \mathbf{x}^T\mathbf{Q}$. See chapter 3 for details.

Subspace model: it consists of the Subspace constraint introduced by Heeger and Jepson [45], interpreted as a *dynamical system*, rather than as an algebraic constraint. $\alpha = V$ is the direction of heading, which is a three-dimensional vector with unit norm. The space of unknown parameters is the sphere of all possible directions of translation: $M = \mathbf{S}^2$. The function h is the orthogonal complement of the range space of a matrix $\mathcal{C}(\mathbf{x}, V)$ of coefficients of the 2-D motion field equation, which depends upon

the image projection of each feature point $\mathbf{x}^i \doteq \pi(\mathbf{X}^i)$ and the direction of heading V. See chapter 5 for details.

Point-fixation model: it arises when the sequence of images is taken while fixating some particular feature-point on the image plane [35]. Such a fixation constraint may be specified simply by considering Essential matrices of the form $\mathbf{Q} = RS^T + vSR$ with $v \in \mathbb{R}_+$ the velocity along the fixation axis and $S \doteq [0\ 0\ 1]\wedge$. The model h remains the same as in the Essential model. See chapter 7 for details.

Point-plus-line fixation model: if, in addition to fixating a point, we impose that another point passes through a given line, we further restrict the parameters to be of the form $\mathbf{Q} = RS^T + vSR, R = e^{[\omega_1\ \omega_2\ 0]\wedge}, v \in \mathbb{R}_+, S = [0\ 0\ 1]\wedge$. See chapter 7 for details.

Plane-plus-parallax model: it describes the residual motion after the image has been warped as to compensate for the motion of a plane [8]. We can impose the plane-fixation constraint simply by restricting the parameters of the Essential model to unit-norm 3×3 matrices of the form $\alpha = T\wedge$, and the parameter space is the two-dimensional unit sphere, as in the Subspace model: $T \in M = \mathbf{S}^2$. See chapter 7 for details.

11.1.2 Formulating the estimation task for the extended models

In the extended models (11.4) derived from the basic constraints of rigidity and perspective, all unknown parameters are *state variables* of the model. Such states evolve in a space that is *not a linear space.* For instance, rotation matrices do not sum up to produce another rotation matrix, and so for unit-norm vectors. Rotation vectors and spherical coordinates are an instance of a system of *local coordinates* on a curved space (such as the set of rotation matrices or the unit-sphere).

The first step in order to make the model (11.4) suitable for designing an EKF that estimates the state from the measurements is to transform the model into local coordinates: to this end we substitute to R its local-coordinate correspondent rotation vector $\Omega_R \in \mathbb{R}^3$, such that [3] $R = e^{\Omega_R\wedge}$. The state of the model becomes $\xi \doteq [\dots\ Z^i\ \dots\ T\ ,\ \Omega_R] \in \mathbb{R}^{N+6}$. We have already assumed that the measurement noise n^i is white, zero-mean and Gaussian and that the motion parameters are described by a random walk, so that the

[3] Note that Ω_R is just an alternative way of representing R and is different from Ω, which represents the instantaneous rotational velocity of the viewer moving frame.

model in local coordinates is driven by a white, zero-mean Gaussian process. In order to avoid *saturation* of the filter (see section 11.1.4), we add a Gaussian noise n_{Z^i} with a small variance also to the first N components of the state model:

$$Z^i(t+1) = R_3(t)Z^i(t) + T_3(t) + n_{Z^i}(t). \tag{11.6}$$

We can proceed in a similar way for the "integral" model (11.4-right), whose state-space is transformed into local coordinates using the exponential map: A small residual noise is added to all components of the state model in order to prevent saturation:

$$\begin{cases} Z_0^i(t+1) = Z_0^i(t) + n_{Z^i}(t) \\ \Omega(t+1) = \Omega(t) + n_\Omega(t) \\ V(t+1) = V(t) + n_V(t) \\ \Omega_R(t+1) = Log_{SE(3)}\left(e^{\Omega(t)\wedge}e^{\Omega_R(t)\wedge}\right) + n_{\Omega_R}(t) \\ T(t+1) = e^{\Omega(t)\wedge}T(t) + V(t) + n_T(t) \\ \mathbf{y}^i(t) = \pi\left(R(t)\mathbf{y}^i(t_0)Z_0^i(t) + T(t)\right) + n^i(t) + n_y^i \end{cases} \tag{11.7}$$

where the last error term in the measurement equation takes into account the error in measuring the coordinates of the projections at the initial time instant $\mathbf{y}^i(t_0)$. The function $Log_{SE(3)}$ indicates the (local) inverse function of the exponential map $R = e^{\Omega_R\wedge}$ (see [81] for details [4]). The variance of the measurement error, Σ_n and Σ_{n_y} can be inferred from the properties of the optical flow/feature tracking algorithm [6]. The variance of the noises that drive the random walk model, Σ_*, with $* = n_{Z^i}, n_\Omega, n_V, n_{\Omega_R}, n_T$ are *tuning parameters*, and must be assigned by the engineer according to some criteria which we will discuss in section 11.1.4. The above model is a local-coordinate version of the *structure-motion-velocity* model described in chapter 2.

The models in eq. (11.4), modified according to (11.6) and (11.7) respectively, are of the general form

$$\begin{cases} \xi(t+1) = f(\xi(t)) + n_\xi(t) \\ \mathbf{y}^i(t) = g(\xi(t)) + n_{\mathbf{y}^i}(t) \quad \forall\, i = 1 \ldots N \end{cases} \tag{11.8}$$

where f and g are locally smooth functions and the unknown parameters are encoded into the state ξ that

[4] A Matlab routine to compute the exponential map and its inverse can be retrieved via anonymous ftp from vision.caltech.edu under `/pub/matlab/vision/rodrigues.m`.

belongs to the linear space $\mathbb{R}^{N+6}$ for (11.4-left) or $\mathbb{R}^{N+6+6}$ for (11.7). Such models are in a form suitable for applying an Extended Kalman Filter, whose equations can be derived from any standard textbook on stochastic filtering, for instance [55]. The only caveat is the *scale factor* ambiguity, which we discuss in section 11.1.6.

11.1.3 Formulating the estimation task for the reduced models

The reduced models (11.5), unlike the extended ones just discussed, are not yet in a form like (11.8) suitable for applying an EKF. In the remainder of this section we are going to outline a method for performing the identification of the class of models (11.5), which is essentially derived from appendix F.

The first step consists in transforming the identification task into a state-estimation task; this is done by postulating some dynamics for the unknown parameters α. In the case when the camera is mounted on a vehicle, or on a robotic arm, we have some dynamic constraints that govern its motion, typically in the form $\alpha(t+1) = f(\alpha(t), n_\alpha(t))$, where f is some smooth function and n_α some unknown input. In the most conservative approach, we may assume that there are some bounds on the acceleration, due to the fact that the relative motion between the camera and the scene is somewhat smooth, so we may write $f(\alpha(t), n_\alpha(t)) = \alpha(t) \oplus n_\alpha(t)$ with the constraint that $n_\alpha(t)$ is (unknown but) small in some norm. We will explain shortly the meaning of the symbol $\oplus$. If a camera is hand-held, or if there is no information on the device that produced the sequence, then we may want to assume a statistical model for the motion parameters, for instance a random walk. The simplest instance of a random walk is a Brownian motion (first order), where $f(\alpha(t), n_\alpha(t)) = \alpha(t) \oplus n_\alpha(t)$ with n_α a white, zero-mean Gaussian process. The choice of the dynamics of the parameters is part of the design process and depends upon the specific application one is targeting. Here we will restrict to first-order random walks just because they are the simplest models and flexible enough to deal with most situations we have encountered:

$$\alpha(t+1) = \alpha(t) \oplus n_\alpha(t) \qquad \alpha(t_0) = \alpha_0 \tag{11.9}$$

where $n_\alpha \in \mathcal{N}(0, \Sigma_n)$. The reader may now wonder what we mean with the symbol $\oplus$. Since the parameters α do not lie on a linear vector space, we cannot simply sum two elements and hope to obtain a point on M. If we want to induce a sum operation we have to map each point into its local-

coordinate correspondent, perform the sum in the local coordinates, and then map the result back onto the original space. If we call $\xi \doteq \psi(\alpha) \in \mathbb{R}^m$ the local-coordinate correspondent of $\alpha \in M$, we have $\oplus : M \times M \longrightarrow M \; ; \; (\alpha_1, \alpha_2) \mapsto \alpha_1 \oplus \alpha_2 \doteq \psi^{-1}(\psi(\alpha_1) + \psi(\alpha_2))$. The symbol $+$ denotes the usual sum on $\mathbb{R}^m$. For instance, if $\alpha = V \in \mathbf{S}^2$ is a unit-norm three-dimensional vector with spherical coordinates θ, ϕ, such that $V(\theta, \phi) \doteq \begin{bmatrix} cos(\theta)cos(\phi) & sin(\theta)cos(\phi) & sin(\phi) \end{bmatrix}^T$ then $V_1 \oplus V_2 \doteq V(\theta_1, \phi_1) \oplus V(\theta_2, \phi_2) = V(\theta_1 + \theta_2, \phi_1 + \phi_2)$, where the last sums are intended modulo 2π.

Equation (11.9), transformed into local coordinates, will be the state of the filter that estimates the parameters α:

$$\xi(t+1) = \xi(t) + n_\xi(t) \tag{11.10}$$

where $\xi \doteq \psi(\alpha)$ and $n_\xi(t) = \psi(n_\alpha(t))$ and $+$ denotes the usual sum in $\mathbb{R}^m$. Now, if we substitute $\mathbf{y}^i - n^i$ for $\mathbf{x}^i$ in the state of the model (11.5), we get

$$h(\mathbf{y}^i(t-1), \alpha(t))\mathbf{y}^i(t) = \tilde{n}^i(t) \qquad \forall \; i = 1 \ldots N, \tag{11.11}$$

where $\tilde{n}^i$ is a noise process induced by n^i. Notice that $\tilde{n}^i$ is *not* a white noise, for it is correlated within one time step. A method for dealing with such a problem is described in appendix F, while in this chapter we will assume that $\tilde{n}^i$ is approximated by a white noise, whose variance is inferred from the variance of n^i and the linearization of h. If we now put together equations (11.9) and (11.11), after assuming that $\tilde{n}^i$ is white, we end up with a dynamic model for the unknown parameters, having an implicit measurement constraint:

$$\begin{cases} \alpha(t+1) = \alpha(t) \oplus n_\alpha(t) & \alpha(t_0) = \alpha_0 \\ h(\mathbf{y}^i(t-1), \alpha(t))\mathbf{y}^i(t) = \tilde{n}^i(t) & \forall \; i = 1 \ldots N, \end{cases} \qquad \alpha \in M \tag{11.12}$$

which has a local-coordinate correspondent

$$\begin{cases} \xi(t+1) = \xi(t) + n_\xi(t) & \xi(t_0) = \xi_0 \\ h(\mathbf{y}^i(t-1), \psi^{-1}(\xi(t)))\mathbf{y}^i(t) = \tilde{n}^i(t) & \forall \; i = 1 \ldots N. \end{cases} \qquad \xi \in \mathbb{R}^M \tag{11.13}$$

The above model is now in a form suitable for applying an EKF in its version for implicit measurement constraints. This can be easily derived from the standard equations of the EFK, after observing that the

variational model about the best estimate of the current trajectory is linear and *explicit*, and the quantity

$$\epsilon^i(t) = h(\mathbf{y}^i(t-1), \psi^{-1}(\hat{\xi}(t+1|t)))\mathbf{y}^i(t) \tag{11.14}$$

plays the role of the *innovation* (the output prediction error [55]) of the filter. A derivation of the equations of the implicit EKF, which are summarized in the next section, can be found in appendix F.

11.1.4 Implementation and tuning

In the previous sections we have seen that both the extended models (11.4) and the reduced models (11.5) can be put in a form that is suitable for designing an observer, which are (11.8) and (11.13) respectively.

If such models were linear and the model and measurement noises were white, zero-mean and Gaussian, the Kalman filter would guarantee that the innovation ϵ be white, zero-mean and have minimum variance. In the case of a nonlinear model, the "whiteness" of the innovation is considered to be a reliable diagnostic of the filter performance, and it may be evaluated using standard statistical tests, for instance Bartlett's Cumulative Periodogram (the integral spectrum of the prediction error).

What are the statistics of the measurement noise in typical vision applications? The feature-correspondence is known up to some uncertainty, summarized in the noise process $\tilde{n}^i$. Such uncertainty comprises both localization noise, which is usually zero-mean and in the order of few pixels standard deviation, and large errors due to mismatches. Such errors are intrinsic in the functioning of feature tracking/optical flow algorithms, which are based upon a local brightness constancy assumption often violated in real-life situations [6]. These errors cannot be eliminated by the optical flow/feature tracking algorithms; indeed, it is responsibility of the methods that use the optical flow/feature tracking in order to estimate 3-D structure and motion to treat properly both sources of errors, by rejecting outlier measurements due to mismatches, and by exploiting the statistics of the localization error and the redundancy in the measurements in order to minimize their effects. When the noise in the measurements is far from white and zero-mean, the statistics of the innovation changes dramatically, which suggests that by doing some simple test on the innovation process we may be able to spot out the outlier measurements due to mismatches in the optical flow/feature tracking. In fact, each component of the innovation measures how consistent each visible feature point is with the current estimate of motion. A test for rejecting outliers based upon such a principle has been proposed in chapter 8.

Therefore, we are going to assume that the measurement noise is white and zero-mean, and we will reject as outliers those feature-points that produce an innovation residual which is not consistent with our statistical model.

We report here, for the sake of completeness, the equations for the Implicit EKF, which can then be applied to the reduced model (11.13), and to the extended model (11.8)

$$\textbf{Prediction step} \begin{cases} \hat{\xi}(t+1|t) = f(\hat{\xi}(t|t)) & \hat{\xi}(0|0) = \hat{\xi}_0 \\ P(t+1|t) = F(t)P(t|t)F^T(t) + \Sigma_\xi(t) & P(0|0) = P_0 \end{cases}$$

$$\textbf{Update step} \begin{cases} \hat{\xi}(t+1|t+1) = \hat{\xi}(t+1|t) + L(t+1)h\left(\mathbf{y}(t-1), \hat{\xi}(t+1|t)\right)\mathbf{y}(t) \\ P(t+1|t+1) = \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + L(t+1)\Sigma_{\tilde{n}}(t+1)L^T(t+1) \end{cases}$$

$$\textbf{Gain:} \begin{cases} L(t+1) = P(t+1|t)C^T(t+1)\Lambda^{-1}(t+1) \\ \Lambda(t+1) = C(t+1)P(t+1|t)C^T(t+1) + \Sigma_{\tilde{n}}(t+1) \\ \Gamma(t+1) = I - L(t+1)C(t+1) \end{cases}$$

Residual variance: $\Sigma_{\tilde{n}}(t+1) = D(t+1)\Sigma_{\tilde{n}}D^T(t+1)$

where $F \doteq \left(\frac{\partial f}{\partial \xi}\right)$, $C \doteq \left(\frac{\partial h}{\partial \xi}\right)$ and $D \doteq \left(\frac{\partial h}{[\partial \mathbf{x}(t)\ \mathbf{x}(t-1)]}\right)$, Σ_* indicates that variance of the process $*$, and P is the variance of the estimation error. In the extended (explicit) models of the form (11.8), we have $h^i(\mathbf{y}(t-1), \xi(t))\mathbf{y}(t) \doteq \mathbf{y}^i(t) - g^i(\xi(t))$; in the reduced models (11.13) we have $f(\xi) = \xi$.

The only ingredients that are needed in order to complete the implementation of the filters are the measurement and model variances $\Sigma_{\tilde{n}}$ and Σ_ξ. For the measurements, we have assumed that the error in the location of each feature-point is independent, with a standard deviation of 1 pixel (0.002 units of focal length in the simulation experiments described in section 11.2), according to the average performance of optical flow/feature tracking techniques [6]. $\Sigma_{\tilde{n}}$ is therefore a $4N \times 4N$ matrix [5] with diagonal elements $4 * 10^{-6}$.

We assume that the model errors n_ξ are uncorrelated, and therefore their variance Σ_ξ is a diagonal matrix. In principle the elements of Σ_ξ corresponding to the structure parameters (in the extended models), and the ones corresponding to Ω_R and T in the integral models should be zero, for the model is *exact*. In order to prevent *saturation* [6] of the filter, we add a noise term whose variance is small relative to the variance of the measurement error (10^{-16}).

The variance of the random walk models for V and Ω is the most crucial to set, for it trades off the "smoothness" of the estimates with the "inertia" of the filter. We have experimented with various types

[5] Note that in the reduced filters we need to keep in memory the measurements at time $t-1$, and the measurement vector is effectively $4N$-dimensional (image-plane coordinates at time t and $t-1$), rather than $2N$-dimensional as in the case of the extended models.

[6] Saturation of the filter can be described as follows: if the variance of the model error is zero, the model is perceived by the filter to be exact, the relative weight of the measurements decreases until the gain becomes zero along some direction, and the filter drifts away without paying attention to the measurements [55].

of motion, and finally set the variance of the random walk parameters to 10^{-6}. This number has nothing magic, and has to be regarded as a reference. In order to be consistent, however, we have maintained the same tuning parameters throughout all the experiments we describe in section 11.2.

11.1.5 Recovering the reduced parameters

The "reduced models" (11.5) are obtained from the extended ones (11.4) via model reduction, as discussed in chapter 7. In essence some of the states are eliminated by solving the measurement equation for such states, and substituted into the model equation. For instance, the Subspace model is obtained by eliminating the depth and rotation parameters from the time-derivative of the measurement equation of the model (11.8).

As a result, the filters based upon the reduced models will only provide an estimate of *some* of the unknown parameters. How can we estimate the remaining ones?

The parameters that are not represented in the state of the reduced models are in a sense "hidden" and can be recovered easily. In fact, we can use the same equation that we solved for *eliminating* them in order to provide an estimate *from the current estimate of the states of the reduced model.* Chapter 5 provides an instance of such an "indirect" estimate for the rotation and structure parameters from the estimated direction of translation.

As for the structure parameters, once motion has been estimated, it can be fed, together with the variance of the estimates, to an algorithm for estimating structure that processes motion error, such as [84, 93]. Structure parameters may also be recovered by simple triangulation (see chapter 3).

11.1.6 Dealing with scale factors

As we have anticipated in previous sections, the structure parameters and the translational velocity are only measurable up to a scale factor which affects the depth of each point and the norm of the relative translation. In fact, it is very well known that an object moving in front of a camera produces the same images as an object which is "twice as far, twice as big and moving twice as fast" [73].

In order to get rid of such an ambiguity we can choose essentially two ways. The first consists in isolating the state variable that corresponds to the scale factor ambiguity and eliminating it. This is done in all reduced filters, where the translational velocity is expressed in spherical coordinates θ, ϕ (azimuth and elevation).

Only the direction of heading, therefore, is estimated while the radius is constant and therefore removed from the state-space.

Alternatively, we may leave the state-space untouched, and saturate the filter along any direction affected by the ambiguity. Note that, by doing so, we are dealing with a model which is globally unobservable, and we just "freeze" our filter onto a slice of the unobservable space. The variance of the model error of any one of the states affected by the ambiguity (for instance the distance of one point in the models (11.4)), is set to zero, and so is the variance of the initial estimate. Each initial condition determines a slice of the state-space which is an observable subset of the state-space. Of course we can observe the trajectory of the model along such slices, but we cannot infer from the measurement in which slice we are.

11.1.7 Integral reduced models

Reduced filters may be implemented in their integral form, simply by referring the structure to the initial time instant and integrating the motion parameters. For instance, in the case of the Essential constraint, the corresponding integral filter is based upon the model

$$\begin{cases} \Omega(t+1) = \Omega(t) + n_\Omega(t) \\ V(t+1) = V(t) + n_V(t) \\ R(t+1) = e^{\Omega(t)\wedge} R(t) \\ T(t+1) = e^{\Omega(t)\wedge} T(t) + V(t) \\ \mathbf{y}^i(t)^T \mathbf{Q}(T(t), R(t)) \mathbf{y}_0^i = \tilde{n}^i(t). \end{cases} \tag{11.15}$$

Here the scale factor may be set by imposing that the initial translation has norm one, by giving it as an initial condition and saturating the initial variance of the estimation error for the norm of translation. This solution, unlike when the scale factor is associated to structure parameters, is very sensitive to drifts since the translational velocity changes in time and therefore the initial guess cannot be updated.

11.1.8 Dealing with occlusions

It must be noticed that, unlike incremental model, all filters based upon an "integral" model (defined relative to the initial time instant) need all the features to be visible throughout the experiment. In the presence of

occlusions and appearance of new features one has to use some ad-hoc heuristics [7]. While all other schemes based upon a first-order random walk estimate *velocity* (or rather relative attitude between successive time instants), the integral filters estimate the attitude of the viewer relative to the initial time instant.

However, we remark that one of the major strengths of the reduced models is that they can integrate motion information over time in absence of continuative tracking of the same point-features, or even using optical flow at a fixed number of locations on the image. In fact, since structure is not represented in the state, we can add and remove features by adding or deleting rows of the measurement equation of the model (11.13), without affecting the continuity of the state. Structure, however, is represented *indirectly* through the innovation process (11.14), whose components are a measure of how consistent each feature is with the current motion interpretation.

11.2 Experiments

We have chosen to use a simulation framework in order to make careful comparisons, since a rigorous *ground truth* is available while the relevant parameters are varied systematically. Such a ground truth is difficult to obtain and impossible to validate for real image sequences.

First, we test the scheme on a real image sequence obtained by rotating a box on top of a chair (the "box sequence", section 11.2.2). Then we build a simulation that mimics the box sequence, and allows us to change the number of visible features, the distance from the viewer, the noise level, the initial conditions for the filters and other structural parameters in a systematic way. The basic setup is described in section 11.2.3, and the following sections outline the results of the experiments. The particular choice of experiment is then validated by testing the algorithms on other motion and structure configurations (section 11.2.11).

11.2.1 Nomenclature

We have implemented a recursive filter for each of the geometric models just summarized. The filter based upon the extended model (11.4-left), which we call the **"structure filter"**, needed very accurate initial conditions for the motion parameters, and therefore it did not converge in most of the situations described in this section. Therefore, the filter for simultaneously estimating structure and motion has been implemented

[7] A technique for dealing with a variable number of features is outlined in [80].

only in its "integral" version, based upon the model (11.4-right), which we call the **"integral structure filter"**. See chapter 2 for details.

We have then implemented the filter derived from the Subspace constraint, called the **"Subspace filter"** in [92], which corresponds to the model (11.12) with the parameter space $M = \mathbf{S}^2$. The velocity of image features is approximated by first differences, and exponential coordinates are used to model the discrete motion between successive time instants. The filter based upon the epipolar constraint of Longuet-Higgins [73] is called the **"Essential filter in local coordinates"** in [90]. These filters are implemented in their incremental version, which can employ both feature tracking or optical flow (velocity vectors at fixed locations on the image-plane) as input. For the sake of comparison with the integral-structure filter, we have also implemented an integral version of the Essential filter, which refers motion to the initial time instant; we call this filter the **"integral Essential filter"**.

We have then implemented one filter for each of the fixation constraints described in chapter 7. The filter derived from fixating a feature-point is called the **"point-fixation filter"**. Similarly, when we fixate a point and a line, we have the **"point-plus-line fixation filter"**, and when we compensate for the motion of a plane we have the **"plane-plus-parallax filter"**, or **"plane-fixation filter"**. All of these filters are obtained from the model (11.12) where, in each case, only the parameter space M changes.

It must be noticed that "integral filters" need all features to be visible throughout the sequence, as opposed to "reduced filters" that can integrate motion information over time even in the presence of features with a very short life-span. Therefore, reduced filters have an advantage in real-life situations, since it is extremely difficult to track single features over long sequences; typically feature-tracking algorithms can trace features over the order of ten frames, and then refresh by selecting a new set of features [6]. In the following sections, however, we are mainly interested in comparing the geometric essence of each scheme, and we have therefore selected all features that survived from the beginning to the end of the experiments, in order to compare integral models against reduced ones.

11.2.2 The basic experiment: the "box sequence"

We report here a test on a sequence of real images that we will later replicate in our simulation environment. This is done mainly for the purpose of motivating the experimental conditions used in the simulations.

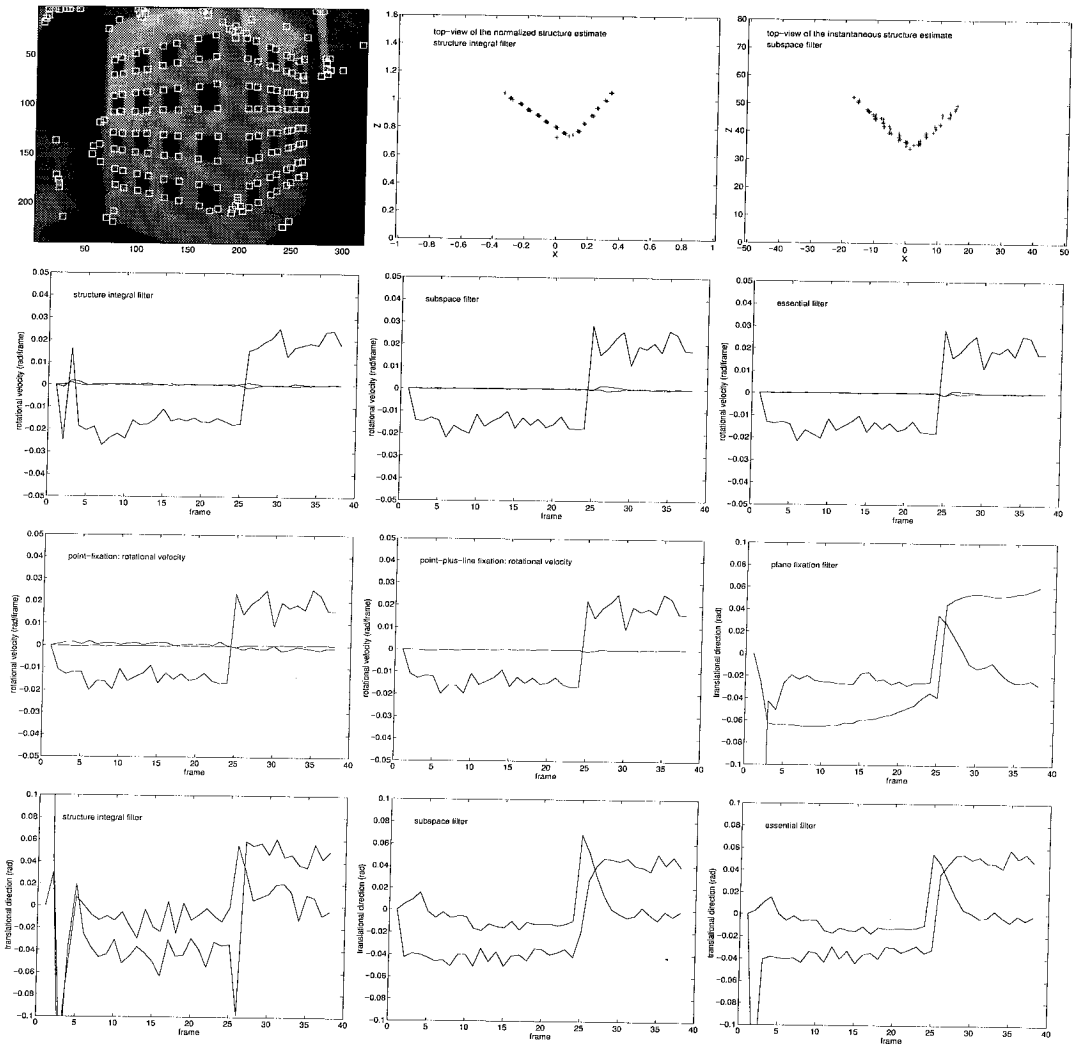

Figure 11.1: *(Row, Column): (1,1) one image of the* **"box sequence"**. *(1,2) normalized structure estimated by the integral structure filter. (1,3) instantaneous estimate of structure by the subspace filter. Rotational velocity estimated by the integral structure filter (2,1), the subspace filter (2,2), the Essential filter (2,3), the point-fixation filter (3,1) and the point-plus-line filter (3,2). The last scheme produces estimates only for two out of the three rotation parameters, since it exploits the fact that the third (cyclorotation) is zero. Direction of translation estimated by the integral structure filter (4,1), the Subspace filter (4,2), the Essential filter (4,3) and the plane-fixation filter (3,3). We plot the two spherical coordinates (azimuth and elevation) as a function of the frame number.*

A box of side approximately $30cm$ is placed on a chair $50cm$ ahead of the camera and rotated by 5 deg/frame circa. The direction of rotation is inverted after 25 frames, and the overall sequence is 40 frames-long.

We have used a multi-scale version of the classical SSD algorithm [75] for tracking a number of features. In order to test the integral filters we have selected only the features that survived from the first to the last frame.

The setting used for each filter is exactly the same used for the simulation experiments which is described in the next sections, and no ad-hoc tuning was performed. Initial conditions were zero for all schemes, and a noise level of one pixel std was hypothesized for the feature tracking.

In figure 11.1 (top row) we show one image of the test sequence (left), with the feature points highlighted, and the estimates of structure performed by the integral structure filter (middle), normalized so as to place the center of mass at unit distance from the viewer. The figure shows a top view of the scene at the initial time instant, and it can be seen that the qualitative structure of the box is estimated correctly. In the right plot we show the instantaneous estimate of structure that comes as a byproduct from the Subspace filter, as discussed in section 11.1.5. Note that such estimate only uses the instantaneous measurements and the current estimate of motion, and is therefore less precise. All other schemes do not provide an estimate of structure *directly*. However, their estimates of motion may be fed to any structure-from-motion module that processes motion error, as done for instance in [84, 93].

In figure 11.1 (rows 2-4) we show the estimates of the rotational velocity and the direction of translation (azimuth and elevation). The plane fixation constraint does not provide an estimate of the rotational velocity *directly*. Similarly, the point-fixation and the point-plus-line fixation constraints do not provide a direct estimate of the direction of translation, but only the translational velocity along the fixation axis.

Of course, in the absence of a ground truth it is only possible to appreciate the qualitative behavior of each estimator. In order to perform a rigorous quantitative evaluation of the properties of each model, it is necessary to employ a simulation platform, which we describe in the next section.

11.2.3 Simulation setup

We have generated a simulation that mimics the box experiment described in the previous section. A cloud of $N = 20$ dots is distributed at random within a cubic volume of side $1m$ at a distance $d = 2m$ from the

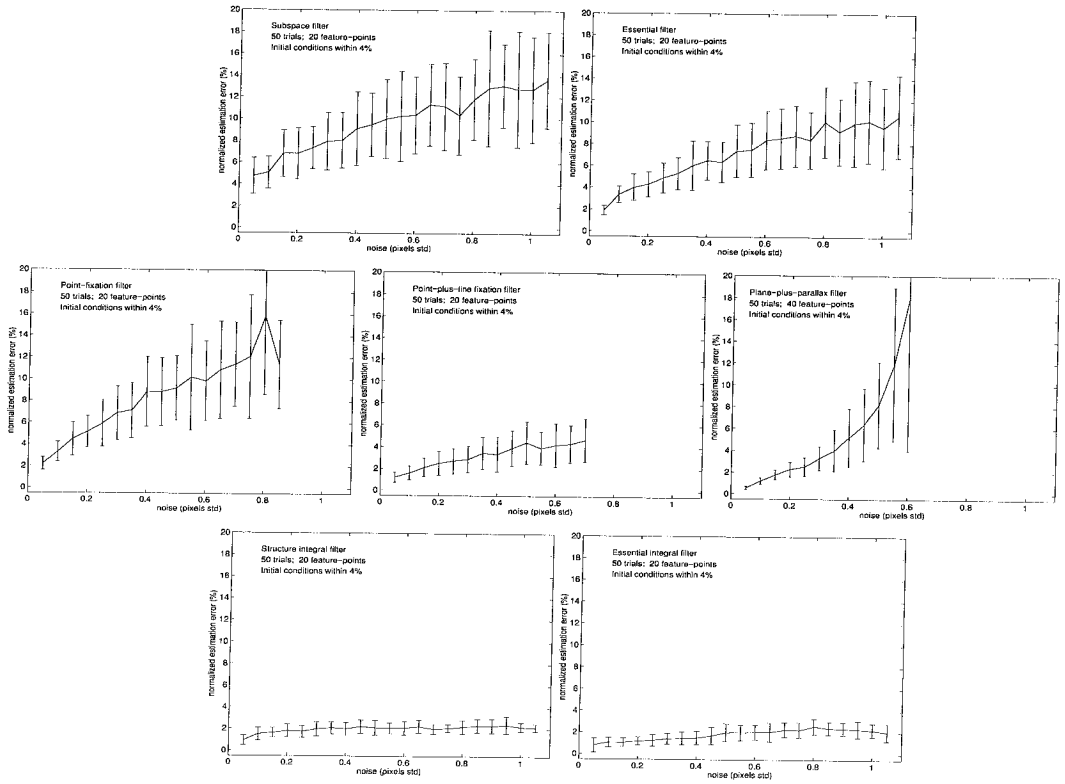

Figure 11.2: **Accuracy experiment.** *50 trials, with 20 feature-points (except for the plane-fixation filter, see also figure 11.6), starting at initial conditions distributed at random within 4% of the true parameters while the noise level increases from* 0.1 *to* 1.1 *pixels std, according to the standard performance of feature tracking algorithms. The scaled norm of the estimation error is plotted against the noise level. The filters enforcing a fixation constraint (middle row), cease converging consistently for less than one pixel noise. Note that integral filters (bottom row) have an advantage in performance, since they can count on an increasingly large baseline. For the integral structure filter we display only the error in the estimates of motion parameters.*

viewer. These dots are projected onto an ideal image plane with unit focal length and 500×500 pixels, corresponding to a visual angle of approximately 30^o and therefore approximately $3.5'$ of visual angle per pixel. White, zero-mean Gaussian noise has been added to the projections with a standard deviation n_0 varying between 0.1 and 12 pixels. The cloud is then rotated about an axis parallel to the image-plane and passing through its center with a constant velocity [8] of 4 deg/frame. The basic experiment is then altered by varying systematically the parameters of the simulation. All tuning parameters remain the same throughout the experiments.

11.2.4 Accuracy

[8] If the reader is not comfortable with this assumption, we suggest a quick look at section 11.2.12.

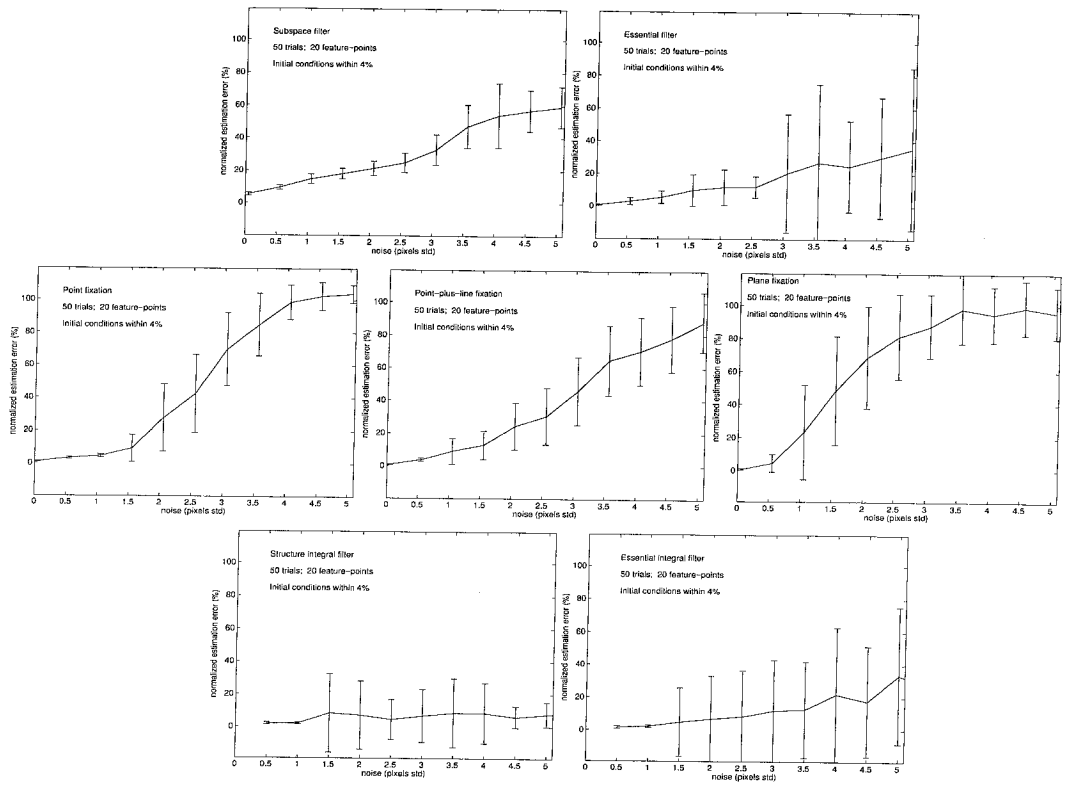

Figure 11.3: **Accuracy/robustness experiment.** *The conditions were the same described in figure 11.2, except that the noise level goes from* 0.1 *to* 5.1 *pixels std and we did not remove the instances when the filters did not converge. The scaled norm of the estimation error is plotted against the noise level after the filters have settled. The size of the error-bars can be considered a measure of robustness, for it indicates the consistency of each filter across trials.*

Each scheme is tested on a sequence containing 20 point-features, with initial conditions distributed normally at random around the true motion parameters, with a standard deviation of 4% of the norm of the true parameters. The noise level is increased from 0.1 to 5.1 pixels std, and the normalized estimation error is evaluated over a window of 10 frames, after the filters have settled (between frame 50 and 60). In figure 11.2 we plot the norm of the estimation error against the noise level for a window between 0.1 and 1.1 pixels, according to the average performance of feature-tracking/optical-flow techniques [6]. In order to evaluate the *accuracy*, we have plotted only the instances when the filters have convergence in all 50 trials. We display the mean error, and visualize the standard deviation using error-bars.

It may be noticed that the Subspace filter does not converge to zero error in the absence of noise and is in general less precise, since it has to cope with the approximation of the derivative of the position of the features on the image-plane using first-differences (upper-left plot). The schemes that impose fixation constraints, either for a point (middle-left), a line (middle-center) or a plane (middle-right) cease converging consistently for noise levels around 0.6 pixels std. This is due to the propagation of the errors in fixating noisy features.

Integral filters (figure 11.2 bottom) can count on an increasingly large baseline, for structure is referred to the initial time-instant and motion is modeled as a second-order random walk, and exhibit therefore a better performance.

In figure 11.3, we plot the norm of the estimation error against the noise level that increases from 0.1 to 5.1 pixels without removing the instances when the filters did not converge. We have performed 50 trials of the experiment, and we display the mean error, and visualize the standard deviation using error-bars. This experiment evaluates a mixture of accuracy and robustness, since the size of the error-bars gives an idea of the consistency of the performance across trials.

11.2.5 Robustness

In this experiment we assess the robustness of each filter, intended as the capability to retain a correct estimate in the presence of increasing noise. We have performed 50 trials, with initial conditions distributed at random within 10% of the true parameters, and we have tested whether the filter has reached convergence after 50 time steps. In order to formulate a convergence verdict we test both the estimation error and

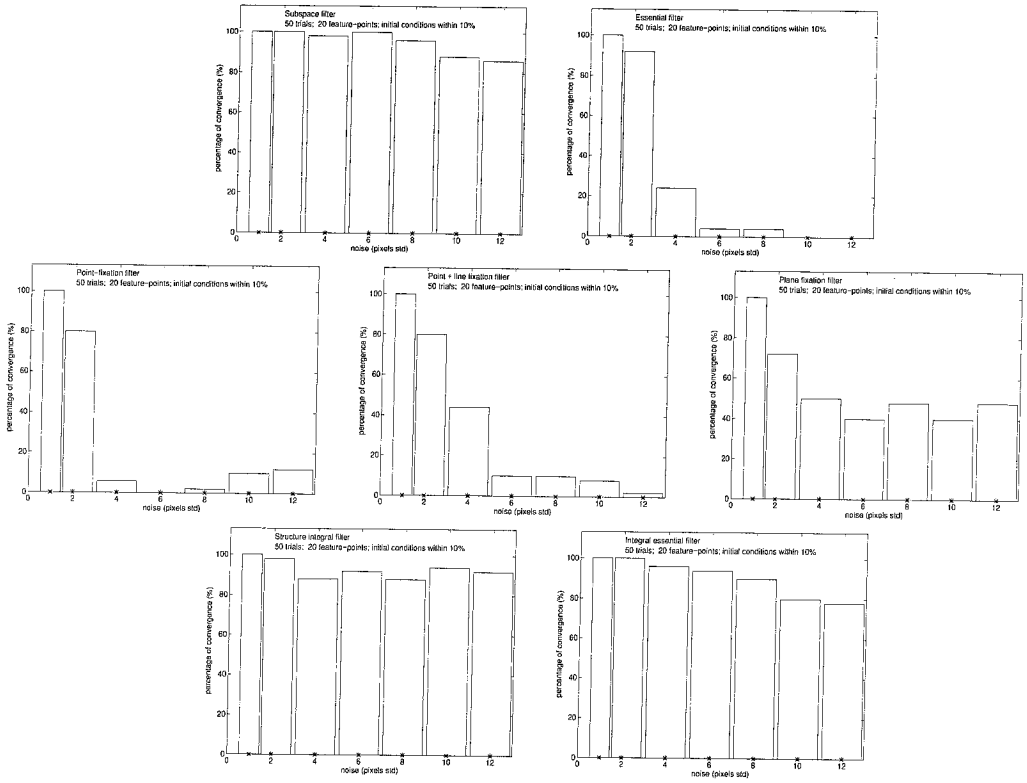

Figure 11.4: **Robustness experiment.** *50 trials with the initial conditions distributed at random within 10% of the true value, and the noise level increased from 1 to 12 pixels std. The histograms represents the percentage of the experiments in which the filters reached convergence. Integral filters (bottom row) exhibit better robustness properties than reduced filters, with the exception of the Subspace filter (top-left).*

the periodogram of the innovation. In fact, the criterion for the filter to be operating correctly is that the innovation be "as white as possible". The periodogram, which is the integral of the prediction error spectrum, is a measure of how "white" the innovation is. However, occasionally filters may get stuck in "local minima" where the innovation is small, but the estimation error is large.

In figure 11.4 we report a histogram of the percentage of trials that have reached convergence as a function of the noise level that ranges between 1 and 12 pixels std. It can be seen that the filters that enforce fixation constraints (middle row) are significantly less robust than the ones based upon explicit reduction. Integral filters (bottom row) are in general more robust than reduced filters, with the exception of the Subspace filter (top-left), which proves remarkably robust.

11.2.6 Convergence

In this experiment we test the convergence properties of each model, by changing the initial conditions at random within a region that grows from 1% to 100% of the true values of the parameters. In figure 11.5 we plot an histogram that counts the percentage of successful convergences as a function of the size of the perturbation of the initial conditions. Noise is half a pixel std.

The filters based upon the fixation assumptions (middle row) have convergence problems, most probably due to the effects of noise propagated through the fixation constraint.

Integral filters (bottom row) prove more sensitive to initial conditions than reduced ones. For the structure integral filter this is due to the observability properties of the model, discussed in [88], while for the Essential integral filter this is most probably due to the mechanism of propagation of scale, which consists in saturating the norm of the initial translational velocity. Such a filter is subject to a drift that increases with perturbations in the initial conditions.

11.2.7 Dependence upon the number of visible points

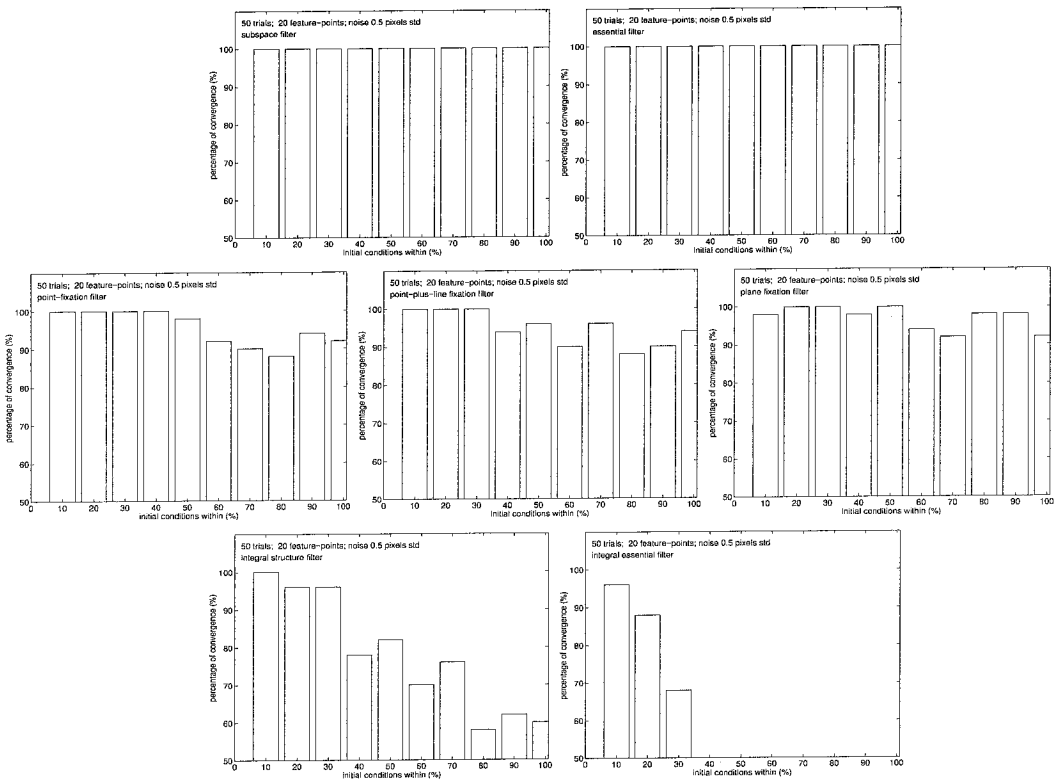

Figure 11.5: **Convergence experiment.** *50 trials with* 0.5 *pixel std error, while the initial conditions are chosen at random with Gaussian distribution with* σ *ranging from* 10% *to* 100% *of the true parameters. Integral filters (bottom row) exhibit decreased robustness relative to reduced filters. For the structure integral filter (bottom-left) this is mainly due to the observability properties of the model having structure in the state, while for the integral Essential filter (bottom-right) this behavior is due to the mechanism of propagation of scale over time.*

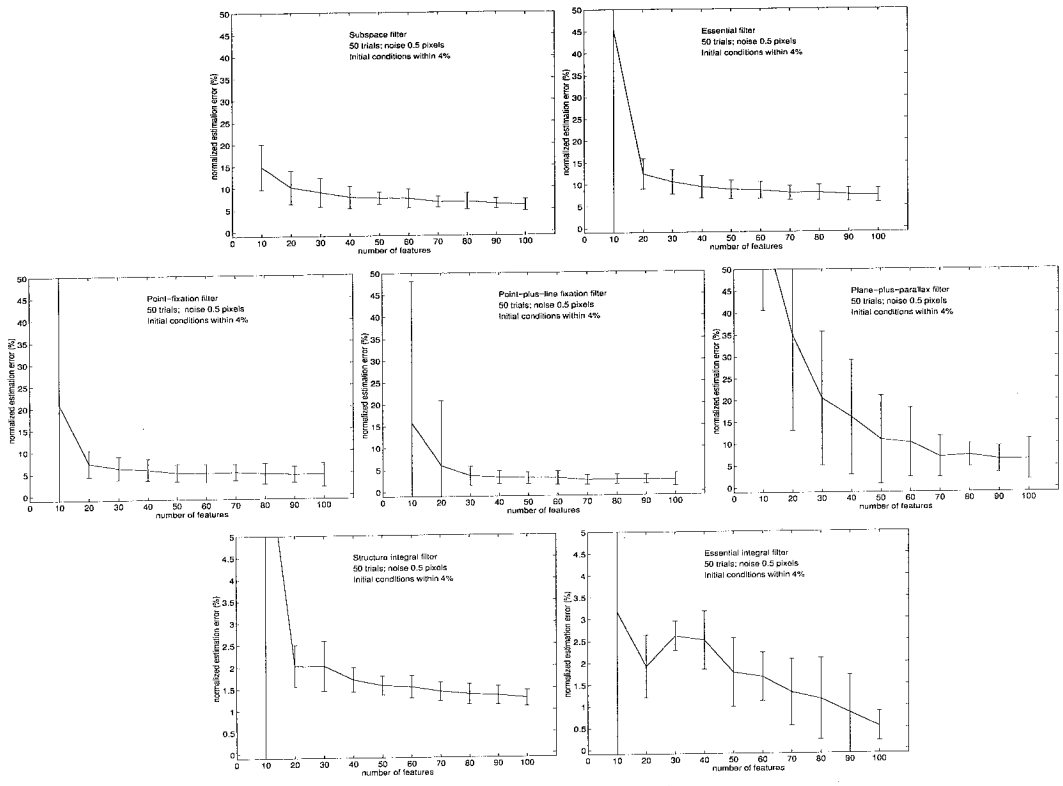

Figure 11.6: **Dependence upon the number of features.** *The norm of the estimation error is plotted against the number of visible features, for a noise level of half a pixel and initial conditions within 4%. The Subspace filter (top-left) has an advantage over other schemes in that it needs fewer features for reaching convergence. However, the computational cost of such a filter is quadratic in the number of features, unlike all other schemes whose complexity is linear. Note that all filters can actually reach convergence in the presence of less than 5 feature-points (for small noise and small acceleration) since motion information is integrated over time. This is an advantage over two-views algorithms that need at least 5 (or 8) features to be visible at all times. Note that the plane-fixation filter needs more features in order to achieve performance similar to other reduced filters. For this reason the accuracy experiment in figure 11.2 has been performed with 20 feature-points for all filters, except for the plane-fixation filter which had 40. Note that the performance improves marginally beyond 50 features.*

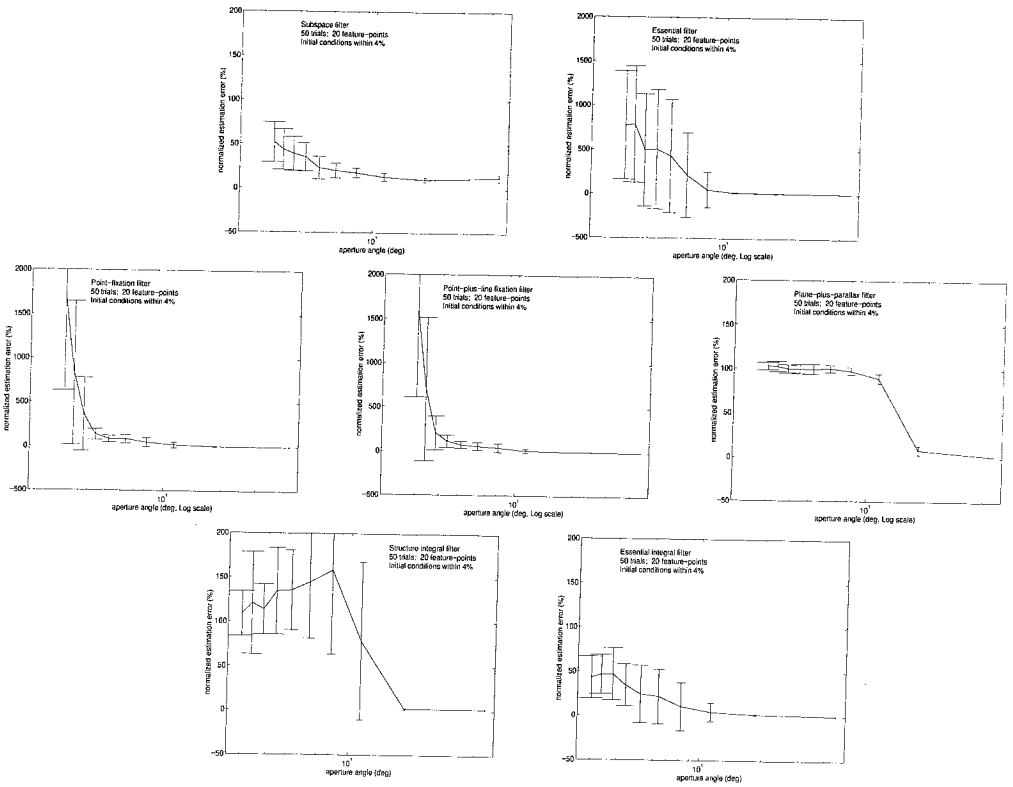

Figure 11.7: **Dependence upon the aperture angle.** *Norm of the estimation error as a function of the aperture angle that ranges from 2^o to 40^o.*

In figure 11.6 we display the norm of the estimation error as a function of the number of features, which range from 10 to 100. In general performance levels at 50 points, for the noise levels and initial conditions considered. An exception is the plane-fixation filter, which needs more points in order to accurately warp the images, and estimate the residual direction of translation. The Subspace filter seems to have an advantage in that it needs fewer points. However, such a filter has a quadratic complexity, and therefore it becomes computationally intensive for more than 70 feature-points.

11.2.8 Dependence upon the aperture angle

All models based upon full perspective projection need a wide field of view in order for the higher-order perspective effects to be appreciable. We have decreased the aperture angle from 40 down to 2 degrees: most filters seem to prefer aperture angles larger than 10 degrees, while the plane-fixation filter and the integral structure filter need at least 20 degrees of visual angle to achieve satisfactory performance (figure 11.7).

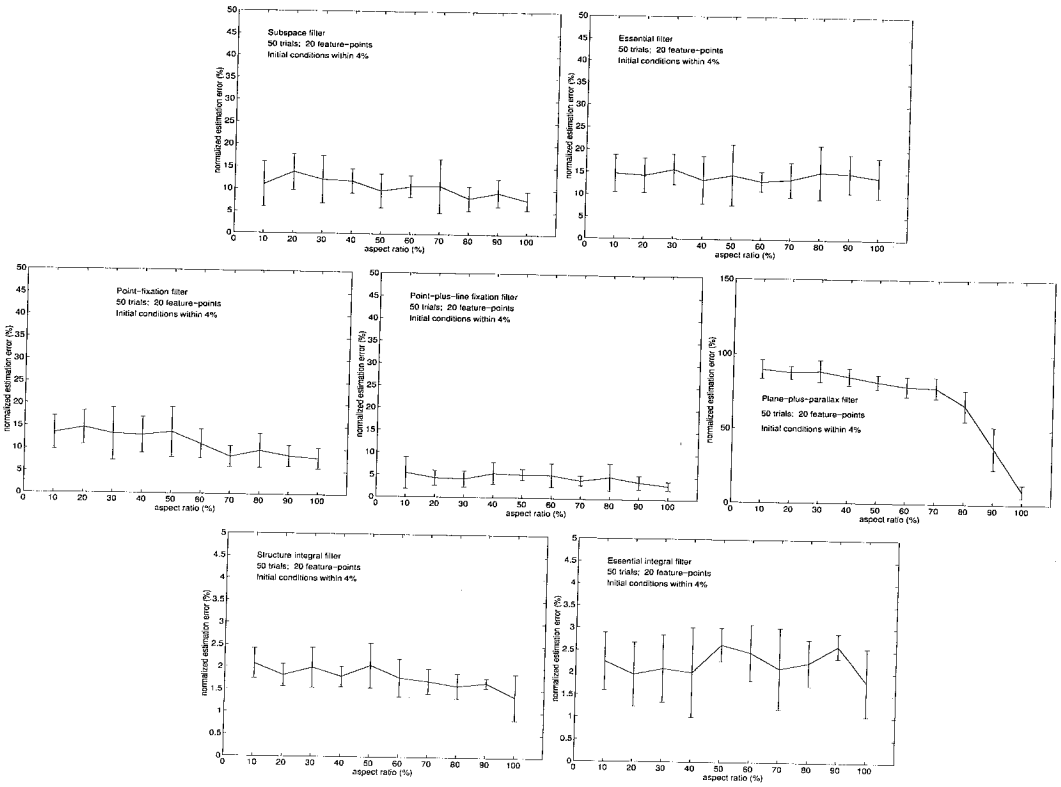

Figure 11.8: **Dependence upon the bas-relief ambiguity.** *The norm of the estimation error is plotted against the "thickness ratio" of the cloud of points being viewed (ratio between width and depth), which ranges between 10% and 100%. The error curve is almost flat for all schemes, except for the plane-fixation filter (middle-right), whose error increases as the scene approaches a plane. When the scene approaches a plane, the warped images have no parallax, and therefore the residual translation has norm zero, and the direction of translation (which is the state of the filter) can be arbitrary without violating the constraints.*

11.2.9 Sensitivity to the "bas-relief" ambiguity

We have taken the original cubic cloud of points, and reduced one of the dimensions to a fraction of the original side, ranging from 100% (cubic cloud) down to 10% (flat cloud). The norm of the estimation error as a function of the "flatness" of the cloud is plotted in figure 11.8. Most filters do not seem to be bothered by such a deformation, for the aperture angle considered (30^o). Notice that one can view such a deformation of the cloud as a reduction of the effective field of view, which is however limited to the times when the cloud shows the thinner face.

An exceptional behavior is exhibited by the plane-fixation filter (middle-right). In fact, the estimation error seems to increase dramatically as the cloud approaches a plane. This, however, does not mean that the filter is not operating correctly. In fact, as the cloud approaches a plane, the warping operation stabilizes

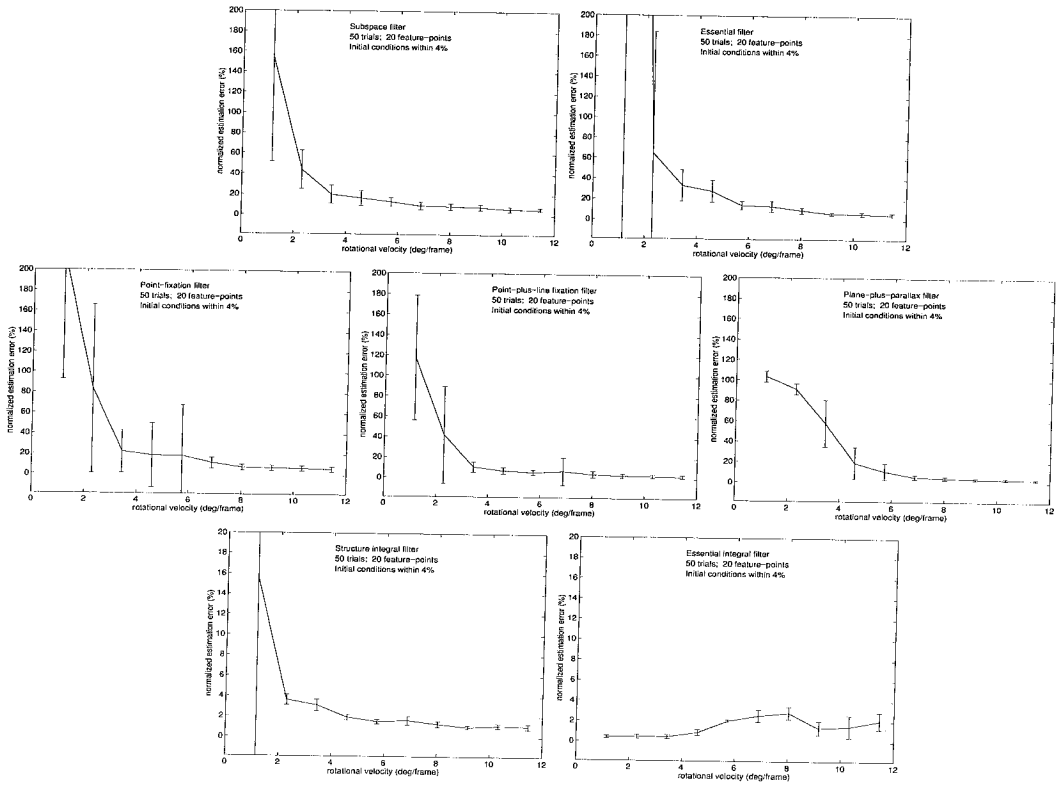

Figure 11.9: **Dependence upon the sampling rate.** *The Subspace filter (top-left), which is based upon a differential model, converges for smaller velocities. In principle its performance should degrade as such velocity increases, since image velocities are approximated by first differences. However, the exponential coordinatization helps maintaining good performance even in the presence of large image-motions. The performance of the integral Essential filter is somewhat odd. Since the filter is based upon a second-order model, and therefore it can count on an increasingly large baseline, it can handle small motions quite well. However, when the instantaneous baseline increases, the bias in the estimate of scale increases, which causes a degradation of the performance.*

such a plane up to the point in which the residual parallax is zero (in the limit of a flat plane). Therefore the norm of the residual translation is zero, and its direction is undetermined.

11.2.10 Dependence upon the parallax (sampling rate)

In the basic experiment the cloud of dots rotates about an axis parallel to the image-plane by 4 degrees per frame. In figure 11.9 we show how performance changes as the rotational velocity varies between 1 and 12 degrees/frame. The Subspace filter is based upon a differential model, and therefore it prefers small rotations. There is, however, a tradeoff between the first-difference approximation of the image-velocity and the amount of parallax in the data. As the velocity increases, the data are better conditioned, but the

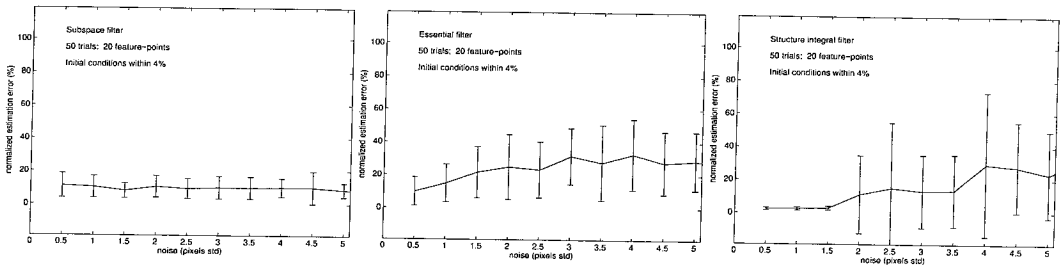

Figure 11.10: **Alternative motions.** *The accuracy/robustness experiment of figure 11.3 is repeated for some alternative motions. In the left plot we display the performance of the Subspace filter for a forward translation of 30 cm/frame. Although the average norm of image-motion vectors is similar to that of the box experiment, the data are less ambiguous, for the effects of rotation and translation do not superimpose. The same motion has been estimated by the Essential filter, and the results are shown in the middle plot. We have also considered translation along a direction parallel to the image-plane by 20 cm/frame. The estimation error for the integral structure filter is reported in the right plot. Compare with figure 11.3 top-left, top-right and bottom-left respectively.*

first-order approximation of the image velocity degrades. The exponential coordinatization of motion helps improving the filter for large image-motions.

The behavior of the Essential integral filter (bottom-right) is almost inverse to the other filters. In fact, it degrades as the image-motion increases. This is most probably due to the mechanism of propagation of scale, which is subject to biases that increase with the size of the image-motion.

11.2.11 Other types of motion

Throughout this section we have considered the "box experiment" as a paradigm. Here we consider other types of motion. In a first experiment we consider forward translation within an infinite cloud of points, where only the ones that fall within a visual angle of 30 degrees are seen. Translation is 30 cm/frame in order to produce an image-motion of size comparable to that of the box sequence. Note that we cannot test integral filters on this sequence, for points move out of the visual field as the viewer translates forward. Results are qualitatively similar to those obtained for the "box experiment". As an example, in figure 11.10 we display the results of the accuracy/robustness experiment for the Essential filter and the Subspace filter. In general this motion is "simpler" than the roto-translational motion of the box experiment, and performance is better.

We have also considered translation along a direction parallel to the image-plane by 20 cm/frame. The scene is the usual cloud of 20 points of side $1m$ at $2m$ from the viewer. As time goes by, the cloud moves farther away, and the effective aperture angle decreases. Nevertheless the performance is comparable with

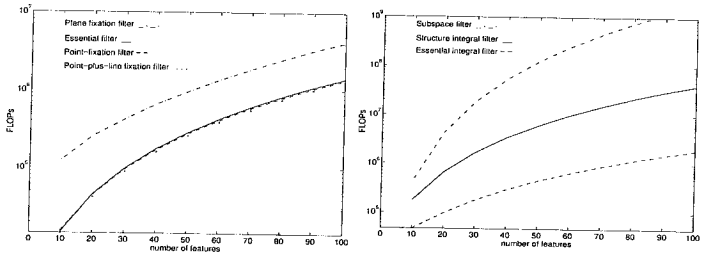

Figure 11.11: **Complexity**: *number of floating point operations as a function of the number of visible features. This count includes the overhead of our Matlab implementation. The Subspace filter has been implemented using a tensor package that does not exploit the sparse structure of the matrices involved in the computation.*

that obtained in the box experiment. In figure 11.10 (right) we show the performance of the structure integral filter.

11.2.12 A remark on "constant velocity" and first-order random walks

In the incremental models we have chosen a first-order random walk to describe the dynamics of the unknown parameters. Integral models can be interpreted as a second-order random walk. The only reason for choosing such random-walk models is that they are a good compromise between simplicity and flexibility. As we have pointed out already, *any other dynamical or statistical model* can be used in place of the first-order walk in any one of the filters described in this chapter, as long as it preserves the observability properties of the overall system. The reader who is uncomfortable with modeling motion as a first-order random walk may consider looking at an experiment described in chapter 5, where the velocity of the cloud of the same synthetic experiment just described is modulated first by a *sinusoid*, then by a *saw-tooth* discontinuous function, and then by a *second-order* random walk.

11.3 Discussion and interpretation of the results

We have compared the various models under controlled conditions, in order to evaluate the geometric properties of each constraint. It emerges that the models obtained by reduction using *fixation*, i.e. using output-dependent changes of coordinates, are in general less effective in all respects: precision, robustness and convergence properties. This is surprising, for one expects that the fewer the degrees of freedom, the better-

conditioned the optimization task should be. Our finding can be explained by the fact that, when reduction is performed using changes of coordinates that depend on the noisy measurements, the effects are propagated in a non-linear fashion across the states of the filter, and even keeping track of the second-order statistics of the errors does not help. "Explicit reduction", on the other hand, does not require use of the measured output, and helps achieve desirable properties such as global observability of the dynamic model [88]. Note that we could reach this conclusion only because the unifying framework allowed us to compare the models that exploit the fixation constraints versus the *same* models based on general motions, simply by changing the geometry of the parameter space while using the same dynamic model and the same estimation technique.

Integral filters are in general more accurate and robust than reduced ones, with the exception of the Subspace filter that proves remarkably insensitive to measurement noise. On the other hand, integral models are more sensitive to perturbations in the initial conditions, due either to the observability properties of the model or to the mechanism of scale propagation.

Other practical aspects, such as the presence of occlusions, need also to be taken into consideration. In fact, in the presence of occlusions, the integral structure filter has a disadvantage over the reduced models that do not include structure parameters in the state, for it has discontinuities in the estimates each time a new feature enters the field of view, or each time a feature disappears. Furthermore, the integral structure filter needs full-fledge feature tracking, and cannot use the optical flow at a fixed number of locations on the image.

The computational load of the schemes proposed are comparable, and range approximately between $40KFlops$ per frame and $10MFlops$ per frame depending upon the scheme, the number of features and the implementation. In figure 11.11 we report the number of floating point-operations as a function of the number of points for our `Matlab` implementation. Such implementation is not optimized and the count includes the overhead from the Matlab server. We feel that each one of the schemes we have tested could be implemented in real-time on standard processors *once the feature tracking/optical flow* is available. Motion and structure estimation are not the crucial bottleneck for real-time systems; feature-tracking/optical flow, on the contrary, is quite demanding and needs to be further optimized in order to run in real-time on low-cost hardware platforms [6].

Chapter 12 What next?

In this thesis we have focused on a low-level point-wise representation of the scene. The geometry of "Structure from Motion" of generic configurations of feature-points is now fairly well understood thanks to the efforts, among others, of Ullmann, Longuet-Higgins, Faugeras, Tomasi and Kanade. The contribution of this thesis is in the direction of processing image information over time, by casting the problem of Structure from Motion within the framework of Dynamical Systems.

Such a point-wise representation is sufficient in a number of applications such as autonomous navigation and robotics manipulation, where a coarse description of the environment suffices to accomplish the tasks at hand. There is, however, a whole range of applications of motion vision for shape representation that demand a more refined description of the environment. For instance, visual data storage (image databases) and transmission (image-sequence compression), virtual reality and multi-media, reverse-engineering of movies, landscape reconstruction for CAD modeling or endoscopic surgery planning, and more in general any system requiring a human to interact with a "computer model" of the environment.

A natural way of refining the representation is to impose *models* on the environment and *group* feature points into higher-level descriptors such as surfaces, solid objects, structures. Then problem-specific information such as a-priori information about the environment can be inserted at the highest level, rather than being enforced early in the data processing, thus jeopardizing the flexibility of the overall system.

In some applications, for instance three-dimensional rendering for Computer Graphics, such a hierarchical representation is built "by hand": first point-wise structure is estimated, then a polygonal mesh is fitted to the three-dimensional data and smoothed. Then different meshes are patched together and rendered. In principle, a system should entail a complete representation of the objects at different levels of resolution and perform the estimation directly on the parameters of the representation, without the need to go through feature-point selection and tracking and then 3-D interpolation. However, as of today, no methods are available for estimating surface structure "directly", and the power of computing hardware is still far from sufficient to accomplish the task in real-time.

There are in the literature some instances of organization of (partial) 3-D information onto surfaces in the field of image compression, for instance the so-called "layers" [2], and "mosaics" [8]. These representations, though primitive in the sense that they involve little of the geometry and none of the dynamics of the environment, have resulted in promising image-compression algorithms. There is also psychophysical [100] and physiological [101] evidence that humans tend to build a representation of their environment by organizing (grouping) depth into surfaces.

Once we enter into this mode of operation, different issues play a crucial role towards a satisfactory solution of the problem, for instance the choice of the model classes and their validation, the segmentation of different models, the design of hierarchical control and decision strategies, communication between different levels of the representation and sensory information fusion.

Bibliography

[1] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, tensor analysis and applications.* Springer Verlag, 2 edition, 1988.

[2] E. Adelson. Layered representations for vision and video. In *Proc. of the IEEE Symposium on Representation of Visual Scenes*, Boston, June 1995.

[3] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 7(4):348–401, 1985.

[4] P. Anandan R. Kumar and K. Hanna. Shape recovery from multiple views: a parallax based approach. *Proc. of the Image Understanding Workshop*, 1994.

[5] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure and focal length. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(6):562–575, 1995.

[6] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. of Computer Vision*, 12(1):43–78, 1994.

[7] A. Basu. Active calibration: anternative strategy and analysis. *Trans. of the IEEE conf. CVPR*, New York, June 1993.

[8] J. Bergen, R. Kumar, P. Anandan, and M. Irani. Representation of scenes from collections of images. In *Proc. of the IEEE Workshop on Visual Scene Representation*, Boston, June 1995.

[9] M. Black. Combining intensity and motion for incremental segmentation and tracking over a long image sequence. *Proc. of the European Conf. on Comp. Vision*, Santa Margherita Ligure, May 1992.

[10] A. Blake, M. Taylor, and A. Cox. Grasping visual simmetry. *Proc. of the Int. Conf. Comp. Vision*, 1993.

[11] W. M. Boothby. *Introduction to differentiable manifolds and riemanniann geometry.* Academic Press, 1986.

[12] J.-Y. Bouguet and P. Perona. Visual navigation using a single camera. *Proc. of the 5 IEEE Int. Conf. Comp. Vision*, Boston, June 1995.

[13] T. E. Boult and L. G. Brown. Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 179–186, October 1991.

[14] P. Bouthemy and E. Francois. Motion segmentation and qualitative scene analysis from an image sequence. *Int. J. Comp. Vision*, 10(2):157–182, 1993.

[15] T. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):90–99, 1986.

[16] R. L. Bryant, S. S. Chern, R. B. Gardner, H. L. Goldshmidt, and P. A. Griffith. *Exterior Differential Systems.* Mathematical Research Institute. Springer Verlag, 1991.

[17] R.S. Bucy. Non-linear filtering theory. *IEEE Trans. Aut. Contr.*, 10, 1965.

[18] P.J. Burt and E.A. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, COM-31:532–540, 1983.

[19] T. K. Carne. The geometry of shape spaces. *Proc. of the London Math. Soc. (3) 61*, 3(61):407–432, 1990.

[20] K. W. Chen. Observability and invertibility of nonlinear systems: a differential algebraic approach. Technical report, Linköping University – Sweden, April 1991.

[21] P. Crouch and F. Silva Leite. The dynamic interpolation problem on riemannian manifolds, lie groups and symmetric spaces. *Technical report*, 1994.

[22] J.L. Crowley, P. Bobet, and C. Schmidt. Maintaining stereo calibration by tracking image points. *Trans. of the IEEE conf. CVPR*, New York, June 1993.

[23] F. Darmon. A recursive method to apply the hough transform to a set of moving objects. *Proc. IEEE, CH 1746 7/82*, 1982.

[24] W. Dayawansa, B. Ghosh, C. Martin, and X. Wang. A necessary and sufficient condition for the perspective observability problem. *Systems and Control Letters*, 25(3):159–166, 1994.

[25] C. Debrunner and N. Ahuja. Motion and structure factorization and segmentation of long multiple motion image sequences. *Proc. of the European Conf. on Comp. Vision*, Santa Margherita Ligure, May 1992.

[26] E. Di-Bernardo, L. Toniutti, R. Frezza, and G. Picci. Stima del moto dell'osservatore e della struttura della scena mediante visione monoculare. *Tesi di Laurea–Università di Padova*, 1993.

[27] E. D. Dickmanns. Historical development of use of dynamical models for the representation of knowledge about real-world processes in machine vision. *Signal Processing*, 35 (3):305–306, 1994.

[28] E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240, 1988.

[29] F. Du and M. Brady. Self-calibration of the intrinsic parameters of cameras for active vision systems. *Trans. of the IEEE conf. CVPR*, New York, June 1993.

[30] O. D. Faugeras. *Three Dimensional Vision, a geometric viewpoint.* MIT Press, 1993.

[31] O. D. Faugeras. On the geometry and algebra of teh point and line correspondences between n images. *Proc. of the Int. Conf. on Comp. Vision*, Boston, June 1995.

[32] O. D. Faugeras, Q.T. Luong, and S. J. Maybank. Camera self-calibration: theory and experiments. *Proc. of the ECCV92, Vol. 588 of LNCS, Springer Verlag*, 1992.

[33] O. D. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. *Proc. of the IEEE Conf. ICCV*, 1987.

[34] O. D. Faugeras and S. J. Maybank. Motion from point matches: multiplicity of solutions. *Int. J. of Computer Vision*, 4(3):225–246, 1990.

[35] C. Fermüller and Y. Aloimonos. Tracking facilitates 3-d motion estimation. *Biological Cybernetics (67), 259-268*, 1992.

[36] D.B. Gennery. Tracking known 3-dimensional object. In *Proc. AAAI 2nd Natl. Conf. Artif. Intell.*, pages 13–17, Pittsburg, PA, 1982.

[37] B. Ghosh, M. Jankovic, and Y. Wu. Perspective problems in systems theory and its application in machine vision. *Journal of Math. Systems Estimation and Control*, 1994.

[38] E.J. Gibson, J. J. Gibson, O. W. Smith, and H. Flock. Motion parallax as a determinant of perceived depth. *J. Exp. Psych. Vol.45*, 1959.

[39] G. Golub and C. Van Loan. *Matrix computations.* Johns Hopkins University Press, 2 edition, 1989.

[40] Guckenheimer and Holmes. *Nonlinear oscillations, dynamical systems and bifurcations of vector fields.* Springer Verlag, 1986.

[41] V. Guillemin and A. Pollack. *Differential Topology.* Prentice-Hall, 1974.

[42] K. Hashimoto, T. Kimoto, T. Ebine and H. Kimura. Image-based dynamic visual servo for a hand-eye manipulator. *Kodama Kimura, editor,* Recent advances in mathematical theory of systems, control, networks, and signal processing II, *pages 609–614. Proceedings of the international symposium of MTNS, Mita Press*, 1991.

[43] K. Hashimoto, T. Kimoto, T. Ebine and H. Kimura. Manipulator control with image-based visual servo. *IEEE Int. Conference on Robotics and Automation*, pages 2267–2272, 1991.

[44] R. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Proc. 2^{nd} Europ. Conf. Comput. Vision, G. Sandini (Ed.), LNCS-Series Vol. 588, Springer-Verlag*, 1992.

[45] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion i: algorithm and implementation. *Int. J. Comp. Vision*, 7(2):95–117, 1992.

[46] J. Heel. Dynamic motion vision. *Proc. of the DARPA image understanding workshop*, 1989.

[47] G. Heinzinger, L. Noakes and B. Paden. Cubic splines on curved spaces. *IMA J. Math. Control and Information*, 1989.

[48] R. Hermann and A. J. Krener. Nonlinear controllability and observability. *IEEE Trans. Aut. Contr.*, 22:728–740, 1977.

[49] C.C. Ho and N.H. McClamrock. Autonomous spacecraft docking using a computer vision systm. *Proc. of the 31st CDC - Tucson, AZ*, 1992.

[50] B. Horn. *Robot vision.* MIT press, 1986.

[51] B.K.P. Horn. Relative orientation. *Int. J. of Computer Vision*, 4:59–78, 1990.

[52] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking of multiple moving objects using temporal integration. *Proc. of the European Conf. on Comp. Vision*, Santa Margherita Ligure, May 1992.

[53] A. Isidori. *Nonlinear Control Systems.* Springer Verlag, 1989.

[54] T. S. Huang J. Weng and N. Ahuja. Motion and structure from two perspective views: algorithms, error analysis and error estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(5):451–476, 1989.

[55] A.H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

[56] A. Jepson and D. Heeger. Linear subspace methods for recovering rigid motion. *Spatial Vision in Humans and Robots, Cambridge University Press*, 1992.

[57] T. Kailath. *Linear Systems.* Prentice Hall, 1980.

[58] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME-Journal of basic engineering.*, 35-45, 1960.

[59] A. Karger and J. Novak. *Space kinematics and Lie groups.* Gordon and Brech Science Publ., 1985.

[60] D. G. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bull. London Math. Soc.*, 16, 1984.

[61] J. J. Koenderink and A. J. Van Doorn. Affine structure from motion. *J. Optic. Soc. Am.*, 8(2):377–385, 1991.

[62] C. Kolb, J. Braun, and P. Perona. Object segmentation and 3d structure from motion. In *Invest. Ophtalmol. Vis. Sci. (Supplement)*, page 1275, 1994.

[63] A. J. Krener and A. Isidori. Linearization by output injection and nonlinear observers. *Systems and Control Letters*, 3, 1983.

[64] A. J. Krener and W. Respondek. Nonlinear observers with linearizable error dynamics. *SIAM J. Control and Optimization*, 1985.

[65] H. Le and D. G. Kendall. The riemannian structure of euclidean shape spaces: a novel environment for statistics. *The Annals of Statistics*, 21(3):1225–1271, 1993.

[66] W. Lee and K. Nam. Observer design for autonomous discrete-time nonlinear systems. *Systems and Control Letters*, 17, 1991.

[67] R.K. Lenz and R.Y. Tsai. Techniques for calibration of the image center and scale factor for high accuracy 3d vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):713–720, 1988.

[68] P. Libermann and C. M. Marle. *Symplectic geometry and analytical mechanics.* Reidel, Dordrecht, 1987.

[69] R. Littlejohn and M. Reinsch. Gauge fields in the separation of rotations and internal motions in the n-body problem. *Rev. Mod. Physics (submitted)*, 1995.

[70] Y. Liu, T.S. Huang and O.D. Faugeras. Determination of camera location from 2d to 3d line and point correspondences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):28–37, 1990.

[71] L. Ljung. *Theory and Practice of Recursive Identification.* MIT press, 1983.

[72] L. Ljung. *System Identification: theory for the user.* Prentice Hall, 1987.

[73] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[74] H.C. Longuet-Higgins. Configurations that defeat the eight-point algorithm. *Mental processes: studies in cognitive science, MIT press*, 1987.

[75] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th Int. Joint Conf. on Art. Intell.*, 1981.

[76] Q.T. Luong and O. D. Faugeras. Determining the fundamental matrix with planes: instability and new algorithms. In *Proc. of the IEEE conf. on Comp. Vision and Patt. Recog.*, New York, June 1993.

[77] K. V. Mardia and I. L. Dryden. Shape distributions for landmark data. *Adv. appl. prob.*, 21(4):742–755, 1989.

[78] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *Int. J. of computer vision*, 3(3):209–236, 1989.

[79] S. J. Maybank. *Theory of reconstruction from image motion.* Springer Verlag, 1992.

[80] P. McLauchlan, I. Reid, and D. Murray. Recursive affine structure and motion from image sequences. *Proc. of the 3rd Eur. Conf. Comp. Vision*, Stockholm, May 1994.

[81] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[82] H. Nijmeijer. Observability of autonomous discrete time nonlinear systems. *Int. J. Control*, 5(36), 1982.

[83] H. Nijmeijer and A. J. van der Schaft. *Nonlinear Dynamical Control Systems.* Springer Verlag, 1990.

[84] J. Oliensis and J. Inigo-Thomas. Recursive multi-frame structure from motion incorporating motion error. *Proc. DARPA Image Understanding Workshop*, 1992.

[85] H. S. Sawhney. Simplifying motion and structure analysis using planar parallax and image warping. *Proc. of the Int. Conf. on Pattern Recognition*, Seattle, June 1994.

[86] J.G. Semple and G.J. Kneebone. *Algebraic Projective Geometry.* Oxford, 1952.

[87] Shizawa. On visual ambiguities due to transparency in motion and stereo. In *Proc. of the Eur. Conf. Comp. Vision*, Santa Margherita Ligure, May 1992.

[88] S. Soatto. Observability/identifiability of rigid motion under perspective projection. In *Proc. of the 33rd IEEE Conf. on Decision and Control*, pages 3235–3240, Dec. 1994.

[89] S. Soatto, R. Frezza, and P. Perona. Motion estimation on the essential manifold. In *Computer Vision, ECCV '94, J.-O. Eklundh (Ed.), LNCS-Series Vol. 801, Springer-Verlag*, pages 61–72, Stockholm, May 1994.

[90] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3):393–414, March 1996.

[91] S. Soatto and P. Perona. Motion from fixation. CDS Technical report CIT-CDS-95-006, California Institute of Technology, February 1995, submitted to ECCV 96.

[92] S. Soatto and P. Perona. Recursive 3-d visual motion estimation using subspace constraints. *Int. J. of Computer Vision*, in press, 1996.

[93] S. Soatto, P. Perona, R. Frezza, and G. Picci. Recursive motion and structure estimation with complete error characterization. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR*, pages 428–433, New York, June 1993.

[94] T. Soderstorm and P. Stoica. *System Identification.* Prentice Hall, 1989.

[95] M. Spivak. *A comprehensive introduction to differential geometry– Voll.I-V.* Publish or perish, 1970-75.

[96] M. A. Taalebinezhaad. Direct recovery of motion and shape in the general case by fixation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(8):847–853, 1992.

[97] W. Thompson, P. Lechleider, and E. Stuck. Detecting moving objects using the rigidity constraint. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(2):162–166, 1993.

[98] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method – 3. detection and tracking of point features. CMU-CS 91-132, School of CS – CMU, April 1991.

[99] P. Torr and D. Murray. Statistical detection of independent movement from a moving camera. *Image and Vision Computing*, 11(4):180–187, 1993.

[100] S. Treue, R. A. Andersen, H. Ando, and E. C. Hildreth. Structure-from-motion: perceptual evidence for surface interpolation. *Vision Research*, 35(1):139–148, 1994.

[101] S. Treue, M. Husain, and R. A. Andersen. Human perception of structure from motion. *Vision Research*, 31(1):59–75, 1991.

[102] S. Ullmann. The interpretation of structure from motion. *Proc. R. Soc. London 203*, 1979.

[103] S. Ullmann. *The Interpretation ov Visual Motion.* MIT Press, 1979.

[104] A. J. Van Der Schaft. Observability and controllability for smooth nonlinear systems. *SIAM J. Control and Optim.*, 20(3), 1982.

[105] H. Von Helmholtz. *Treatise on Physiological Optics.* 1910.

[106] J. Weber and A. Crishnaswami. Planar navigation. Cns-tr, California Institute of Technology, 1996.

[107] J. Weber and J. Malik. Rigid body segmentation and shape description from optical flow. *U.C. Berkeley Technical Report*, 1994.

[108] J. Weng, N. Ahuja, and T. Huang. Optimal motion and structure estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:864–884, 1993.

[109] J. Weng, T.S. Huang, and N. Ahuja. Motion and structure from line correspondences: closed-form solution, uniqueness and optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(3):318–336, 1992.

[110] Z. Zhang and O. Faugeras. *3D dynamic scene analysis*, volume 27 of *Information Sciences.* Springer-Verlag, 1992.

[111] Z. Zhang and O. D. Faugeras. Three dimensional motion computation and object segmentation in a long sequence of stereo frames. *Int. J. of Computer Vision*, 7(3):211–241, 1992.

Part III

Appendices and Background Material

Appendix A Feature tracking

In this chapter we describe a method to automatically select and track point-features in a sequence of images. A point-feature is *defined* as a point that can easily be recognized from one frame to the other (the so-called "correspondence problem"). The basic constraint used to solve (locally in space and time) the correspondence problem is the "brightness constancy" of the region of image surrounding a feature point.

Many algorithms have been proposed to perform this task; for a review see [6]. We will concentrate on a well-known algorithm presented by Lucas and Kanade [75] and then refined by Tomasi and Kanade [98]. It relies upon a differential technique, which is therefore effective only when the spatial and temporal sampling rates are high enough, so that the displacement across frames is small. When this assumption is not satisfied it is possible to apply the same technique in a coarse-to-fine manner, as we discuss.

The material presented in this chapter is not central to the thesis: we treat feature-tracking and camera-calibration as a "front-end" for algorithms performing three-dimensional structure and motion estimation. Therefore, this chapter has been relegated in the appendix, having assumed that we have a method for identifying a number of features on an image and find their correspondent in subsequent images.

A.1 Feature points on an image

Features are often referred to as characteristic points which are representative of the scene being viewed and easily recognizable from one image to the other. In order to develop a technique to automatically extract features we need a more operative definition. First of all it is not easy to identify single points (pixels), so we associate to each point with coordinates (x, y) on an image I a small neighborhood (or window) $\mathcal{W}(x, y)$.

Remark A.1.1 *If the feature-window has constant brightness, then it looks like a homogeneous patch and it cannot be localized in a different image. If the window has a brightness gradient, then it is possible to localize its correspondent on a different frame only* in the direction of the gradient, *since an image-shift normal to the gradient does not modify the brightness pattern of the window patch (this is a manifestation*

of the well-known "aperture problem"*). Therefore, in order to be able to solve the correspondence problem, a window needs to have a significant gradient along two independent direction. This concept is formalized as follows.*

We say that the window $\mathcal{W}$ is a *reliable window* if the integral of the squared spatial gradient of the image is above some threshold τ along two independent directions: if $I(x, y, t)$ is the brightness of the image at the point (x, y) at time t, and $\nabla I(x, y, t)$ is the spatial gradient calculated at that point, then the window being reliable can be expressed as

$$\sigma_{min}\left(\int_{\mathcal{W}(x_0,y_0)} \nabla I \nabla I^T dxdy\right) > \tau, \tag{A.1}$$

where σ_{min} denotes the smallest singular value. In the simplest implementations the partial derivatives are approximated by first differences and the integral by a finite sum. The expression in parenthesis is referred to as *Sum of Squared Difference* (SSD) [75]. We say that the point (x_0, y_0) is a feature point if $\mathcal{W}(x_0, y_0)$ is a reliable window.

After this definition, it is easy to devise an algorithm to extract features:

- set a threshold
- for all points in the image compute the SSD
- if the minimum singular value of the SSD is bigger than the threshold then mark the point as a feature point.

Remark A.1.2 *There are some practical issues that must be addressed in constructing an effective feature selector. For instance, we must avoid selecting different feature-points within the same window, since this could cause feature mismatch during the tracking.*

To this end, one can sort the pixels in an image based upon the value of the smallest singular value of the SSD (which can be taken as a "quality measure" for that point as a potential feature). Then within a given area one may select only the "best" candidate as a feature point.

Optical flow and feature-tracking

Instead of selecting a number of point-features and then trying to estimate their displacement across frames, we may choose a fixed set of locations in the image, and then estimate the velocity of the brightness patches

at those locations, which is what is called *"optical flow"*. Of course in such a case we have to make sure that the brightness patch at each preset location satisfies the SSD criterion, otherwise the optical flow is subject to the aperture problem. Once this is done, the algorithm for estimating feature displacements that we describe in the next section can be used to compute optical flow as well.

The literature on optical flow is quite vast and we do not attempt to give a thorough coverage here. For more details see [6] and references therein.

A.2 SSD algorithm for feature displacement

Let $I(x, y, t)$ be an image, the displacement $\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \doteq d$ is implicitly defined by the following "brightness constancy equation"

$$\frac{d}{dt} I(x, y, t) = I|_{(x,y,t)} - \left(\frac{\partial I}{\partial x}_{|_{(x,y,t)}} \right) d_1 - \left(\frac{\partial I}{\partial y}_{|_{(x,y,t)}} \right) d_2 - I_t|_{(x,y,t)} = 0, \tag{A.2}$$

where $dt = 1$. The above can be rewritten, indicating with ∇I the spatial gradient of the image, as

$$\nabla I|_{(x,y,t)} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = (I - I_t)_{|_{(x,y,t)}}. \tag{A.3}$$

Note that this in fact does not define a unique displacement, since it is a scalar equation in two unknowns. However, we can augment this to a (rank-one) vector equation by multiplying on the right by ∇I^T,

$$\nabla I^T \nabla I_{|_{(x,y,t)}} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \nabla I^T (I - I_t)_{|_{(x,y,t)}} \tag{A.4}$$

and then integrate it on a small window $\mathcal{W}(x, y)$:

$$\int_{\mathcal{W}(x,y)} \nabla I^T \nabla I_{(u,v,t)} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} dudv = \int_{\mathcal{W}(x,y)} \nabla I^T (I - I_t)_{(u,v,t)} dudv. \tag{A.5}$$

If we assume that the displacement vector is constant at each point of the window $\mathcal{W}$, we can write the above equation as

$$G \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = e \tag{A.6}$$

where $G = \int_{\mathcal{W}(x,y)} \nabla I^T \nabla I_{(u,v,t)} du dv$ and $e = \int_{\mathcal{W}(x,y)} \nabla I^T (I - I_t)_{(u,v,t)} du dv$. In order to be able to estimate the feature displacement, we must be able to invert the matrix G, which is exactly what we defined to be the SSD in the previous section. Therefore, if we select feature points based upon the SSD criterion, we are guaranteed to overcome the aperture problem, and we can solve for the displacement via

$$\begin{bmatrix} \hat{d_1} \\ d_2 \end{bmatrix} = G^{-1} e. \tag{A.7}$$

A.3 Sub-pixel iteration

In order to achieve sub-pixel accuracy we may iterate the procedure just described in the following way:

- $\hat{d}^0 = G^{-1} e$
- $\hat{d}^{i+1} = G^{-1} e^{i+1}$

where we define

- $e^0 \doteq e$
- $e^{i+1} \doteq \int_{\mathcal{W}} \nabla I^T (I - I_t)_{(u+d_1^i, v+d_2^i, t)} du dv.$

At each step $(u + d_1^i, v + d_2^i)$ is in general not on the pixel grid, so that it is necessary to interpolate the brightness values to obtain image intensity at that location.

A.4 Multi-scale pyramid

One problem common to all differential techniques is that they fail as the displacement across frames is bigger than a few pixels. One possible way to overcome this inconvenience is to use a coarse-to-fine strategy:

- build a pyramid of images by smoothing ans subsampling the original images (see for instance [18])

- select features at the desired level of definition and then propagate the selection up the pyramid
- track the features at the coarser level
- propagate the displacement to finer resolutions and use that displacement as an initial step for the sub-pixel iteration described in the previous section.

Remark A.4.1 *The whole procedure can be very slow for a full-resolution image if implemented on conventional hardware. However, it is highly parallelizable, so that the image could be sub-divided into regions which are then assigned to different processors. In many applications, where a prediction of the displacement of each feature is available, it is possible to process only restricted regions of interest within the image.*

A.5 Uncertainty Analysis

Optical flow/feature displacement is implicitly defined by equation (A.5). In general we do not have direct access to the image I, but only to a sampled and quantized version of it, corrupted by noise. Furthermore, in practical implementations, derivatives are approximated by first difference operators and the integral is a finite sum. However, it is useful to postulate the existence of a continuous and differentiable process I underlying the observations in order to discuss the incremental effect of different approximations a posteriori.

In the following we assume that the image is corrupted by a simple additive noise model, so that the actual observation $\tilde{I}$ is the direct superposition of the "true" image I and some noise process $n_0(x, y, t)$. It is customary to assume that the process n_0 is statistically uncorrelated in space and time, and has a Gaussian distribution. This might be partially motivated by the fact that n_0 is the result of a large number of independent events and hence there is enough substance to invoke central limit theorems. However, a more honest standpoint would be to postulate whiteness and Gaussianity and then try to validate the hypotheses with a detailed statistical analysis. The quantization error, though deterministic in nature, can be modeled as random noise with uniform distribution with domain equal to the quantization step. As a further simplification, we include the quantization error in the additive term n_0 by approximating its uniform distribution by a Gaussian one. In the following analysis we assume for simplicity that n_0 is the collection of all of the noise terms, modeled as a Gaussian white noise with variance σ^2; n_0 is therefore equal to σn, where n is a unit-variance, white, zero-mean Gaussian noise.

Let us now expand (A.5) in a Taylor series around $\sigma = 0$, and calculate the covariance of the displacement measurements up to first-order (the mean is obviously zero).

$$\tilde{I} = I + \sigma n \Rightarrow \nabla\tilde{I} = \nabla I + \sigma\nabla n \tag{A.8}$$

$$\nabla\tilde{I}^T\nabla\tilde{I} \simeq \nabla I^T\nabla I + \sigma(\nabla n^T\nabla I + \nabla I^T\nabla n) + \mathcal{O}(\sigma^2) \tag{A.9}$$

$$\sum_{i,j} \nabla\tilde{I}^T\nabla\tilde{I} \simeq \sum_{i,j} \left(\nabla I^T\nabla I + \sigma\nabla n^T\nabla I + \sigma\nabla I^T\nabla n\right) \ \forall i,j \in \mathcal{W} \tag{A.10}$$

where higher order terms have been neglected. Now call

- $G_1 = \sum_{i,j} \nabla\tilde{I}^T\nabla\tilde{I}$
- $e_1 = \sum_{i,j} \nabla n(I - I_t)$
- $e_2 = \sum_{i,j} \nabla I n_t$
- $\tilde{d} = d + d_n$.

Then we can rewrite (A.5) as

$$\tilde{G}\begin{bmatrix} \tilde{d_1} \\ d_2 \end{bmatrix} = \tilde{e} \tag{A.11}$$

or, after expanding in Taylor series around $\sigma = 0$,

$$Gd + \sigma G_1 d + \sigma G d_n = e + e_1 + e_2 \tag{A.12}$$

where higher order terms have been neglected. Then by definition of d we have $Gd = e$, hence

$$G_1 d + G d_n = e_1 + e_2 \Rightarrow d_n = G^{-1}(e_1 + e_2). \tag{A.13}$$

Our goal now is to find an expression for the covariance of d_n. Since the matrix G_1 is linear in n, we can write it as a linear operator L_G times a vector obtained by stacking the elements $n_{i,j} \forall i,j \in \mathcal{W}^+$ on top of each other. $\mathcal{W}^+$ is a window augmented to allow the calculation of the spatial gradient of n. e_1 can also be

written as a linear operator L_{e_1} times the same vector. e_2, on the other hand, can be written as a linear operator L_{e_2} whose size is twice that of e_1, since it must multiply all the noise components in the current window as well as those in the previous one in order to calculate the time difference. Hence the size of L_G and L_{e_1} is $2 \times (w+1)^2$, while L_{e_2} is $2 \times 2(w+1)^2$, where w is the size of the original window $\mathcal{W}$. The above steps can be summarized as

$$G_1 d = L_G \begin{bmatrix} n_{1,1,t} \\ n_{1,2,t} \\ \vdots \\ n_{w+1,w+1,t} \end{bmatrix} \tag{A.14}$$

$$e_1 = L_{e_1} \begin{bmatrix} n_{1,1,t} \\ n_{1,2,t} \\ \vdots \\ n_{w+1,w+1,t} \end{bmatrix} \tag{A.15}$$

$$e_2 = L_{e_2} \begin{bmatrix} n_{1,1,t} \\ n_{1,2,t} \\ \vdots \\ n_{w+1,w+1,t} \\ n_{1,1,t+1} \\ n_{1,2,t+1} \\ \vdots \\ n_{w+1,w+1,t+1} \end{bmatrix} \tag{A.16}$$

and the generic row element of the operator L_G is:

$$\begin{bmatrix} \cdots & (2I_{x_{i,j}} d_1 + I_{y_{i,j}} d_2) & \cdots & (-2I_{x_{i,j}} d_1 - I_{x_{i,j}} d_2 - I_{y_{i,j}} d_2) & \cdots & (I_{x_{i,j}} d_2) & \cdots \\ \cdots & (I_{y_{i,j}} d_1) & \cdots & (-2I_{y_{i,j}} d_2 - I_{y_{i,j}} d_1 - I_{x_{i,j}} d_1) & \cdots & (2I_{y_{i,j}} d_2 + I_{x_{i,j}} d_1) & \cdots \end{bmatrix} \tag{A.17}$$

which multiplies the generic error term

$$\begin{bmatrix} \vdots \\ n_{i+1,j} \\ \vdots \\ n_{i,j} \\ \vdots \\ n_{i,j+1} \\ \vdots \end{bmatrix} \tag{A.18}$$

while the generic row element of the operator L_{e1} is

$$\begin{bmatrix} \cdots & (I_{i,j} - I_{t_{i,j}}) & \cdots & -(I_{i,j} - I_{t_{i,j}}) & \cdots & 0 & \cdots \\ \cdots & 0 & \cdots & -(I_{i,j} - I_{t_{i,j}}) & \cdots & (I_{i,j} - I_{t_{i,j}}) & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ n_{i+1,j} \\ \vdots \\ n_{i,j} \\ \vdots \\ n_{i,j+1} \\ \vdots \end{bmatrix} \tag{A.19}$$

and, finally, the generic row term of the operator L_{e2} is

$$\begin{bmatrix} 0 & \cdots & -I_{x_{i,j}} & \cdots & 0 & \cdots & | & \cdots & 0 & \cdots & I_{x_{i,j}} & \cdots & 0 & \cdots \\ 0 & \cdots & -I_{y_{i,j}} & \cdots & 0 & \cdots & | & \cdots & 0 & \cdots & I_{y_{i,j}} & \cdots & 0 & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ n_{i+1,j,t} \\ \vdots \\ n_{i,j,t} \\ \vdots \\ n_{i,j+1,t} \\ \vdots \\ n_{i+1,j,t+1} \\ \vdots \\ n_{i,j,t+1} \\ \vdots \\ n_{i,j+1,t+1} \\ \vdots \end{bmatrix}. \tag{A.20}$$

After reordering the terms $n_{i,j,t}$ in the above definitions we can write

$$d_n = G^{-1} \left((L_G + L_{e_1}) \begin{bmatrix} n_{1,1,t} \\ n_{1,2,t} \\ \vdots \\ n_{w+1,w+1,t} \end{bmatrix} + L_{e_2} \begin{bmatrix} n_{1,1,t} \\ n_{1,2,t} \\ \vdots \\ n_{w+1,w+1,t} \end{bmatrix} \right) \tag{A.21}$$

Call the ordered error vectors $n_1 \doteq \begin{bmatrix} \vdots \\ n_{i+1,j} \\ \vdots \\ n_{i,j} \\ \vdots \\ n_{i,j+1} \\ \vdots \end{bmatrix}$ and $n_2 \doteq \begin{bmatrix} \vdots \\ n_{i+1,j,t} \\ \vdots \\ n_{i,j,t} \\ \vdots \\ n_{i,j+1,t} \\ \vdots \\ n_{i+1,j,t+1} \\ \vdots \\ n_{i,j,t+1} \\ \vdots \\ n_{i,j+1,t+1} \\ \vdots \end{bmatrix}$, then we can calculate the variance of the above expression:

$$\Sigma_{d_n} = E\left[d_n d_n^T\right] \tag{A.22}$$

$$G^{-1}(L_G + L_{e1})E\left[n_1 n_1^T\right](L_G + L_{e1})^T {G^{-1}}^T + G^{-1}L_{e2}E\left[n_2 n_2^T\right]L_{e2}^T {G^{-1}}^T \tag{A.23}$$

since the noise processes are supposed to be uncorrelated in space and time, the variance of the long column vectors will be identity matrices of the appropriate sizes, so that we have finally

$$\Sigma_{d_n} = G^{-1}\left(L_G L_G^T + L_G L_{e_1}^T + L_{e_1} L_G^T + L_{e_1} L_{e_1}^T + L_{e_2} L_{e_2}^T\right) {G^{-1}}^T. \tag{A.24}$$

Appendix B Camera calibration

Feature selection and tracking provides us with the pixel coordinates (row-column) of a number of points in the images across time. The process of establishing a correspondence between such pixel coordinates and metric coordinates is called *calibration*. In this section we only describe the problem at a superficial level, enough for what is treated in subsequent chapters. For a thorough analysis of the problem of calibration, see the book of Faugeras [30] (pages 33-66) and references therein.

In this section we use the symbols (x, y) to denote the metric coordinates, while $(\mathbf{i}, \mathbf{j})$ are the row-column coordinates as outcomes of the feature-tracking step.

B.1 Perspective projection, camera reference and pixel coordinates

Suppose that we are looking at a three-dimensional scene through a camera, and call $\mathbf{X} \in \mathbb{R}^3$ the coordinates of the generic point in space, relative to an orthonormal reference frame centered in the center of projection. The surface of the CCD sensor could be modeled as a plane of equation $\mathbf{X}_3 = 1$, so that the third axis of our reference frame coincides with the optical axis, and the image-plane is parallel to the first two reference axes. In such a case, the coordinates of the projection of the point $\mathbf{X}$ can be written as an ideal perspective (central) projection

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \doteq \pi(\mathbf{X}) = \begin{bmatrix} \frac{\mathbf{X}_1}{\mathbf{X}_3} \\ \\ \frac{\mathbf{X}_1}{\mathbf{X}_3} \end{bmatrix}. \tag{B.1}$$

The coordinates $\mathbf{x}$ described position on the image plane relative to a reference frame centered in the *optical center* (the intersection of the image-plane with the optical axis), with the axes aligned with the first two axes of the reference in 3-D. Such a choice, which is what we call the "camera reference", is made on purpose for the perspective projection to have the simple form above.

In order to establish metric correspondences between the features and the three-dimensional world, it is necessary to find the change of coordinates between the camera reference and the pixel coordinates. We will model such a change of coordinates as an affine transformation of the plane, and a quadratic correction that compensates for the radial distortion due to the lens. The affine transformation can be written as follows:

$$\begin{bmatrix} \mathbf{i} \\ \mathbf{j} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_0 \\ \mathbf{j}_0 \end{bmatrix} + A \begin{bmatrix} x \\ y \end{bmatrix} \tag{B.2}$$

where $(\mathbf{i}, \mathbf{j})$ are the pixel coordinates (row, column) of a given point, $(\mathbf{i}_0, \mathbf{j}_0)$ is the location of the optical center expressed in pixel coordinates, and A is a generic 2×2 matrix. In order to simplify the model we could assume that A is diagonal, which amounts to assuming that the pixels are rectangular and not "diamond-shaped". Such an assumption is often reasonable, so we will simply consider

$$A = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}. \tag{B.3}$$

The scalars s_1, s_2 can be interpreted as the size of the pixels along the two principal directions expressed in terms of focal lengths. If we further assume that the pixels are actually square, then $s_1 = s_2$ is the focal length. Such an assumption, however, is often violated on commercial cameras.

Once we have changed the reference from row-column to camera coordinates, we need to compensate for the radial distorsion introduced by the lens. We model such a distorsion as a transformation

$$\mathbf{x} \mapsto \tilde{\mathbf{x}} = \mathbf{x}(1 + k\|\mathbf{x}\|^2). \tag{B.4}$$

Therefore, calibrating a camera amounts to recovering the set of parameters $\mathbf{i}_0, \mathbf{j}_0, s_1, s_2, k$, which correspond to the optical center, the pixel size and the radial distorsion.

B.2 Recovering camera parameters

Let us call $\mathcal{K}$ the overall transformation from pixel coordinates $(\mathbf{i},\mathbf{j})$ to distorted camera coordinates $\tilde{\mathbf{x}}$:

$$\mathcal{K} : \mathbb{R}^2 \to \mathbb{R}^2; \tilde{\mathbf{x}} \mapsto (\mathbf{i},\mathbf{j}). \tag{B.5}$$

We call $\mathcal{K}^{-1}$ its inverse function. If we write the perspective projection of a number of points $\mathbf{X}^i \; \forall \, i = 1 \ldots N$,

$$\tilde{\mathbf{x}}^i = \mathcal{K}^{-1}(\mathbf{i},\mathbf{j}) = \pi(\mathbf{X}^i) \; \forall \, i = 1 \ldots N \tag{B.6}$$

and isolate the pixel coordinates

$$(\mathbf{i},\mathbf{j}) = \mathcal{K}\left(\pi(\mathbf{X}^i)\right) \; \forall \, i = 1 \ldots N \tag{B.7}$$

we see that they depend upon the camera parameters $\mathbf{i}_0, \mathbf{j}_0, s_1, s_2, k$, which are present in the function $\mathcal{K}$.

In order to recover such camera parameters, one could *measure* the coordinates of a number of points in three-dimensions $\mathbf{X} = [\mathbf{X}^1, \ldots \mathbf{X}^N]$, and the pixel-coordinates of their corresponding projections $\mathbf{i},\mathbf{j}$, and then set up an optimization problem of the form

$$\mathbf{i}_0, \mathbf{j}_0, s_1, s_2, k = \arg\min \| (\mathbf{i},\mathbf{j}) - \mathcal{K}_{\mathbf{i}_0,\mathbf{j}_0,s_1,s_2,k}\left(\pi(\mathbf{X})\right) \|. \tag{B.8}$$

To this end, it is customary to use a *calibration rig*, which is a pattern of points that are easily detected as feature points, whose coordinates are known with high precision relative to some reference frame. Figure B.1 displays an image of a typical calibration rig.

However, the coordinates of such points are usually known in a reference frame which is *not the camera frame*, since it is very difficult to know the exact location in space of the optical center. Therefore, we do not measure the three-dimensional coordinates $\mathbf{X}^i$ but, rather, the coordinates relative to a "world reference", $\mathbf{X}^i_w$. The change of coordinates between the camera and the world reference can be described as

$$\mathbf{X}^i = R\mathbf{X}^i_w + T \tag{B.9}$$

where T is the (unknown) location of the center of projection of the camera in the world reference, and R is

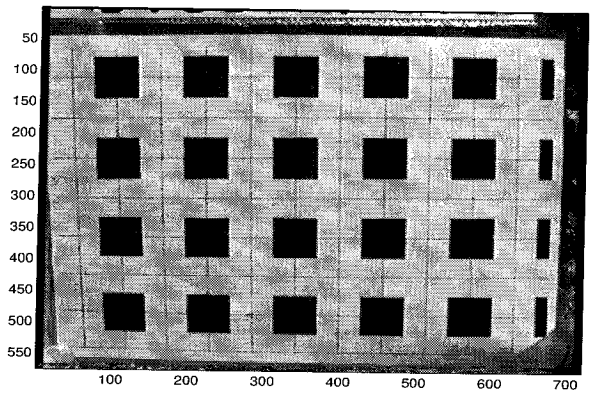

Figure B.1: *An image of a calibration rig. The coordinates of the corners of the checkerboard pattern are precisely measured relative to the center of the rig. Their corresponding projection is measured on the image-plane in terms of row-column coordinates. The calibration process exploits these measurements in order to recover the intrinsic and extrinsic parameters of the imaging device.*

the (unknown) orientation of the camera frame relative to the world reference. (T, R) are often referred to as *extrinsic calibration parameters*, as opposed to the *internal parameters* that describe the location of the optical center, pixel size and radial distorsion.

Then our optimization problem must be solved with respect to all intrinsic and extrinsic parameters:

$$\mathbf{i}_0, \mathbf{j}_0, s_1, s_2, k, T, R = \arg\min \|(\mathbf{i}, \mathbf{j}) - \mathcal{K}_{\mathbf{i}_0, \mathbf{j}_0, s_1, s_2, k}\left(\pi(R\mathbf{X}_w + T)\right)\|. \tag{B.10}$$

Note that in the above optimization the parameters in the matrix R are *not free*, since they must preserve the structure of the rotation matrix which in the above expression is denoted by saying that R must belong to the space of special (unit-determinant) orthonormal matrices $SO(3)$.

Remark B.2.1 *One may set up an optimization routine to estimate the local coordinates of the parameters iteratively from the above constraints. There are of course issues concerning the presence of local minima and the initialization of the iteration, which we will not pursue here. See [50] (chapter 13) for more details.*

Remark B.2.2 *In order to be able to recover all intrinsic and extrinsic parameters, the data must be "sufficiently exciting". For instance, if the calibration pattern is parallel to the image-plane, the location of the optical center cannot be distinguished from the translation component of the extrinsic parameters. We will not get into the details of the analysis of singular calibration rigs, for which the reader is referred to [30].*

For our purposes it will suffice to know that a planar rig slanted with respect to the image plane can be used to find the calibration parameters after having performed an iterative minimization on equation (B.10) in a least-squares sense. To avoid falling into local minima one may initialize the iteration at different initial conditions, and then check the consistency of the estimates.

Appendix C Linear maps, Gram-Schmidt and the Singular Value Decomposition

We assume that the reader is familiar with the basic concepts of linear algebra.

C.1 Linear maps and linear groups

A *linear transformation* of a linear (vector) space (modeled as $\mathbb{R}^n$) is defined as a map $T : \mathbb{R}^n \to \mathbb{R}^n$ such that

- $T(x+y) = T(x) + T(y)\ \forall\ x, y \in \mathbb{R}^n$
- $T(\alpha x) = \alpha T(x)\ \forall\ x \in \mathbb{R}^n, \alpha \in \mathbb{R}$.

If we consider the ring of all $n \times n$ matrices over the field $\mathbb{R}$, its group of units $\mathcal{GL}(n)$ – which consists of all $n \times n$ *invertible* matrices and is called the *general linear group* – can be identified with the set of linear maps:

$$T : \mathbb{R}^n \to \mathbb{R}^n;\ x \mapsto T(x) = \mathbf{T}x \mid \mathbf{T} \in \mathcal{GL}(n). \tag{C.1}$$

We recall that a set G is a *group* if it closed with respect to an operation, call it $\cdot$

$$\begin{aligned} \cdot : G \times G &\longrightarrow G \\ (g_1, g_2) &\mapsto g_1 \cdot g_2 \end{aligned} \tag{C.2}$$

which is associative, has a null element and an inverse:

1. $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)\ \forall\ g_1, g_2, g_3 \in G$ (associative)
2. $\exists\ e \in G \mid g \cdot e = g\ \forall g \in G$ (null element)
3. $\forall\ g \in G\ \exists\ g^{-1} \in G \mid g \cdot g^{-1} = g^{-1} \cdot g = e$ (inverse).

The set of $n \times n$ non-singular matrices is a group under the usual matrix product. Such a group can also be identified with the *metric* (vector) space $\mathbb{R}^{n^2}$.

We say that a linear transformation of a space with inner product is *orthogonal* if it preserves such inner product:

$$< \mathbf{T}x, \mathbf{T}y >=< x, y > \ \forall\ x, y \in \mathbb{R}^n. \tag{C.3}$$

The set of $n \times n$ orthogonal matrices forms the *orthogonal group* $O(n)$. If M is a matrix representative of an orthogonal transformation, expressed relative to an orthonormal reference frame, then it is easy to see that the orthogonal group is characterized as

$$O(n) = \{M \in \mathcal{GL}(n) \mid MM^T = I\}. \tag{C.4}$$

The determinant of an orthogonal matrix can be ± 1. The subgroup of $O(n)$ with unit determinant is called the *special orthogonal group* $SO(n)$.

C.2 Gram-Schmidt orthonormalization

A matrix in $\mathcal{GL}(n)$ has n independent rows (columns). A matrix in $O(n)$ has orthonormal rows (columns). The Gram-Schmidt procedure can be viewed as a map between $\mathcal{GL}(n)$ and $O(n)$, for it transforms a non-singular matrix into an orthonormal one. Call $\mathcal{L}_+(n)$ the subset of $\mathcal{GL}(n)$ consisting of lower triangular matrices with positive elements along the diagonal. Such matrices form a subgroup of $\mathcal{GL}(n)$. Then we have

Theorem C.2.1 *(Gram-Schmidt)* $\forall M \in \mathcal{GL}(n)\ \exists! L \in \mathcal{L}_+(n)\ E \in O(n)$ *such that*

$$M = LE \tag{C.5}$$

Proof:

The proof consists in constructing L and E iteratively from the rows $\mathbf{m}_{i.}$ of M:

$$\begin{array}{rcccl} \mathbf{v}_{1.} & \doteq & \mathbf{m}_{1.} & \longrightarrow & \mathbf{e}_{1.} \doteq \frac{\mathbf{v}_{1.}}{\|\mathbf{v}_{1.}\|} \\ \mathbf{v}_{2.} & \doteq & \mathbf{m}_{2.} - < \mathbf{m}_{2.}, \mathbf{e}_{1.} > \mathbf{e}_{1.} & \longrightarrow & \mathbf{e}_{2.} \doteq \frac{\mathbf{v}_{2.}}{\|\mathbf{v}_{2.}\|} \\ \vdots & \doteq & \vdots & \longrightarrow & \vdots \\ \mathbf{v}_{n.} & \doteq & \mathbf{m}_{n.} - \sum_{i=1}^{n-1} < \mathbf{m}_{i.}, \mathbf{e}_{i.} > \mathbf{e}_{i.} & \longrightarrow & \mathbf{e}_{n.} \doteq \frac{\mathbf{v}_{n.}}{\|\mathbf{v}_{n.}\|} \end{array}$$

Then $E = [\mathbf{e}_{1.}^T \dots \mathbf{e}_{n.}^T]^T$ and the matrix L is obtained as

$$L = \begin{bmatrix} \|\mathbf{v}_{1.}\| & 0 & \dots & 0 \\ < \mathbf{m}_{2.}, \mathbf{e}_{1.} > & \|\mathbf{v}_{2.}\| & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \dots & \|\mathbf{v}_{n.}\| \end{bmatrix}$$

Remark C.2.1 *Gram-Schmidt's procedure has the peculiarity of being* causal, *in the sense that the k-th column of the transformed matrix depends only upon rows with index $l \leq k$ of the original matrix. The choice of the name E for the orthogonal matrix above is not random. In fact we will view the Kalman filter as a way to perform a Gram-Schmidt orthonormalization on a peculiar Hilbert space, and the outcome E of the procedure is traditionally called the* innovation.

C.3 Symmetric matrices

Definition C.3.1 *$Q \in \mathbb{R}^{n \times n}$ is* symmetric *iff $Q^T = Q$.*

Theorem C.3.1 *Q is symmetric then*

1. *Let (v, λ) be eigenvalue-eigenvector pairs. If $\lambda_i \neq \lambda_j$ then $v_i \perp v_j$, i.e. eigenvectors corresponding to distinct eigenvalues are orthogonal.*

2. *$\exists$ n orthonormal eigenvectors of Q, which form a basis for $\mathbb{R}^n$.*

3. *$Q \geq 0$ iff $\lambda_i \geq 0 \forall i = 1 : n$, i.e. Q is positive semi-definite iff all eigenvalues are non-negative.*

4. *if $Q \geq 0$ and $\lambda_1 \geq \lambda_2 \cdots \lambda_n$ then $\max_{\|x\|_2=1} < x, Qx >= \lambda_1$ and $\min_{\|x\|_2=1} < x, Qx >= \lambda_n$.*

Remark C.3.1

- *from point (3) of the previous theorem we see that if* $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$ *is the matrix of all the eigenvectors, and* $\Lambda = diag\{\lambda_1 \cdots \lambda_n\}$ *is the diagonal matrix of the corresponding eigenvalues, then we can write* $Q = V\Lambda V^T$*; note that* V *is orthonormal.*

- *Proofs of the above claims are easy exercises.*

Definition C.3.2 *Let* $A \in \mathbb{R}^{m\times n}$*, then we define the induced 2-norm of* A *as an operator between* $\mathbb{R}^n$ *and* $\mathbb{R}^m$ *as*

$$\|A\| \doteq \max_{\|x\|_2=1} \|Ax\|_2^2 = \max_{\|x\|_2=1} < x, A^TAx > .$$

Remark C.3.2

- *Similarly other induced operator norms on* A *can be defined starting from different norms on the domain and co-domain spaces on which* A *operates.*

- *let* A *be as above, then* A^TA *is clearly symmetric and positive semi-definite, so it can be diagonalized by a orthogonal matrix* V*. The eigenvalues, being non-negative, can be written as* σ_i^2*. By ordering the columns of* V *so that the eigenvaluematrix* Λ *has decreasing eigenvalues on the diagonal, we see, from point (e) of previous theorem, that* $A^TA = Vdiag\{\sigma_1^2 \cdots \sigma_n^2\}V^T$ *and* $\|A\|_2 = \sigma_1$*.*

C.4 Structure induced by a linear map

- Let A be an operator from a vector space E over the field of $\mathbb{F}$ (e.g. $\mathbb{R}$) to a space $(F, \mathbb{F})$

- Let E have a scalar product $< , >_E$: $E \times E \longrightarrow \mathbb{F}$ and F have finite dimension and a scalar product $< , >_F$: $F \times F \longrightarrow \mathbb{F}$

- Let the adjont operator A^* be defined by

$$< Ax, y >_F = < x, A^*y >_E \quad \forall x \in E, y \in F$$

of course $A^* : F \longrightarrow E$. If $A \in \mathbb{R}^{m\times n}$, $A^* = A^T$

- Let E be decomposed as:

$$E = Nu(A) \overset{\perp}{\oplus} Nu(A)^{\perp}$$

- Let F be decomposed as $E = Ra(A) \overset{\perp}{\oplus} Ra(A)^{\perp}$.

Theorem C.4.1 *Let A, E, F be defined as above; then*

a) $Nu(A)^{\perp} = Ra(A^*)$

b) $Ra(A)^{\perp} = Nu(A^*)$

c) $Nu(A^*) = Nu(AA^*)$

d) $Ra(A)^{\perp} = Ra(AA^*)$.

C.5 The Singular Value Decomposition (SVD)

The SVD is a useful tool to capture essential features of a linear operator, such as the rank, Range space, Null space, induced norm etc. and to "generalize" the concept of "eigenvalue- eigenvector" pair.

The computation of the SVD is numerically well-conditioned, so it makes sense to try to solve some typical linear problems as matrix inversions, calculation of rank, best 2-norm approximations, projections and fixed-rank approximations, in terms of the SVD of the operator.

C.5.1 Algebraic derivation

Theorem C.5.1 *Let $A \in \mathbb{R}^{m\times n}$ have rank p. Furthermore suppose, WLOG, that $m \geq n$, then*

- $\exists U \in \mathbb{R}^{m\times p}$ *whose columns are orthonormal*
- $\exists V \in \mathbb{R}^{n\times p}$ *whose columns are orthonormal*
- $\exists \Sigma \in \mathbb{R}^{p\times p}, \Sigma = diag\{\sigma_1 \cdots \sigma_p\}$ *diagonal with* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p$

such that $A = U\Sigma V^*$.

Constructive proof

- compute A^*A: it is hermitian and positive semi-definite of dimension $n \times n$. Then order its eigenvalues in decreasing order and call them $\sigma_1^2 \geq \cdots \geq \sigma_p^2 \geq \cdots \sigma_n^2 \geq 0$. Call the σ_i *singular values.*
- From an orthonormal set of eigenvectors of A^*A create an orthonormal basis for $\mathbb{R}^n$ such that $span\{v_1 \cdots v_p\} = Ra(A^*)$ and $span\{v_{p+1} \cdots v_n\} = Nu(A)$. Note that the latter eigenvectors correspond to the zero singular values, since $Nu(A^*A) = Nu(A)$.
- define u_i such that $Av_i = \sigma_i u_i \forall i = 1 : p$, and see that the set $\{u_i\}$ is orthonormal (proof left as exercise).
- Complete the basis $\{u_i\}_{\forall i=1:p}$, which spans $Ra(A)$ (by construction), to all $\mathbb{R}^m$.
- then $A[v_1 \cdots v_n] = [u_1 \cdots u_m] \begin{bmatrix} \sigma_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \sigma_2 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \sigma_p & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \vdots & \vdots & 0_n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & 0_m \end{bmatrix}$ which we name $A\tilde{V} = \tilde{U}\tilde{\Sigma}$
- hence $A = \tilde{U}\tilde{\Sigma}\tilde{V}^T$

Then the claim follows by deleting the columns of $\tilde{U}$ and the rows of $\tilde{V}^T$ which multiply the zero singular values.

C.5.2 Geometric interpretation

Theorem C.5.2 *Let $A \in \mathbb{R}^{n \times n} = U\Sigma V^T$, then A maps $B(0,1) \doteq \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$ to an ellipsoid with half-axes $\sigma_i u_i$*

Proof:

let x, y be such that $Ax = y$. $\{v_1 \cdots v_n\}$ is an orthonormal basis for $\mathbb{R}^n$. with respect to such basis x has coordinates $[< v_1, x >, < v_2, x >, \cdots, < v_n, x >]$. Idem for $\{u_i\}$. Let $y = \sum_{i=1}^n y_i u_i \rightarrow Ax = \sum_{i=1}^n \sigma_i u_i v_i^T x = \sum_{i=1}^n \sigma_i u_i < v_i, x >= \sum_{i=1}^n y_i u_i = y$. Hence $\sigma_i < v_i, x >= y_i$. Now $\|x\|_2^2 = \sum_{i=1}^n < v_i, x >^2 = 1 \forall x \in B(0,1)$, from which we conclude $\sum_{i=1}^n \frac{y_i^2}{\sigma_i^2} = 1$, which represents the equation of an ellipsoid with half-axes of length σ_i.

C.5.3 Some properties of the SVD

Rank and Null space

Theorem C.5.3 *Let$A = U\Sigma V^T$ have rank r; then*

- $Nu(A) = \text{span}\{v_{r+1} \ldots v_n\}$
- $Ra(A^*) = Nu(A)^\perp = \text{span}\{v_1 \ldots v_r\}$
- $Ra(A) = \text{span}\{u_1 \ldots u_r\}$
- $Ra(A)^\perp = Nu(A^*) = \text{span}\{u_{r+1} \ldots u_n\}$

proof: by construction.

Generalized (Moore-Penrose) Inverse

The problems involving orthogonal projections onto invariant subspaces of A, as Linear Least Squares (LLSE) or Minimum Energy problems, are easily solved using the SVD.

Definition C.5.1 *Let $A \in \mathbb{R}^{m \times n}$, $A = U\Lambda V^T$ where Λ is the diagonal matrix with diagonal elements $(\lambda_1, \ldots \lambda_r, 0 \ldots 0)$; then*

$$A^\dagger = U\Lambda_{(r)}^{-1} V^T, \quad \Lambda_{(r)}^{-1} = diag(\lambda_1^{-1}, \ldots \lambda_r^{-1}, 0 \ldots 0)$$

Theorem C.5.4

- $AA^\dagger A = A$
- $A^\dagger AA^\dagger = A^\dagger$

Least squares solution of a linear systems

Theorem C.5.5 *Consider the problem $Ax = b$ with $A \in \mathbb{R}^{m\times n}$ of rank $p \leq min(m,n)$, then the solution $\hat{x}$ that minimizes $\|A\hat{x} - b\|$ is given by $\hat{x} = A^{\dagger}b$.*

Fixed rank approximations

One of the most important properties of the SVD has to deal with fixed-rank approximations of a given operator. Given A as an operator from a space X to a space Y of rank n, we want to find an operator B from the same spaces such that it has rank $p < n$ fixed and $\|A - B\|_F$ is minimal, where the F indicates the Frobenius norm (in this context it is the sum of the singular values).

If we had the usual 2-norm and we calculate the SVD of $A = U\Sigma V^T$, then by simply setting all the singular values but the first p to zero, we have an operator $B \doteq U\Sigma_{(p)}V^T$, where $\Sigma_{(p)}$ denotes a matrix obtained from Σ by setting to zero the elements on the diagonal after the p^{th}, which has exactly the same two norm of A and satisfies the requirement on the rank.

It is not difficult to see the following result

Theorem C.5.6 *Let A, B be defined as above, then $\|A - B\|_F = \sigma_{p+1}$. Furthermore such norm is the minimum achievable.*

Proof: easy exercise; follows directly from the orthogonal projection theorem and the properties of the SVD given above.

Perturbations

Consider a non-singular matrix $A \in \mathbb{R}^{n\times n}$ (if A is singular substitute its inverse by the moore-penrose pseudo-inverse). Let δA be a full-rank perturbation. Then

- $|\sigma_k(A + \delta A) - \sigma_k(A)| \leq \sigma_1(\delta A)\ \forall k = 1 : n$
- $\sigma_n(A\delta A) \geq \sigma_n(A)\sigma_n(\delta A)$
- $\sigma_1(A^{-1}) = \frac{1}{\sigma_n(A)}$

Condition number

consider again the problem $Ax = b$, and consider a "perturbed" full rank problem $(a + \delta A)x = b + \delta b$. Since $Ax = b$, then to first order approximation $\delta x = -A^{\dagger}\delta A x$. Hence $\|\delta x\| \leq \|A^{\dagger}\|\|\delta A\|\|x\|$, from which $\frac{\|\delta x\|}{\|x\|} = \|A^{\dagger}\|\|A\|\frac{\|\delta A\|}{\|A\|} \doteq k(A)\frac{\|\delta A\|}{\|A\|}$. "$k(A)$" is called the condition number of A. It easy to see that $k(A) = \frac{\sigma_1}{\sigma_n}$.

Appendix D Manifolds, tangent spaces, vector fields

D.1 Smooth manifolds

Although an intuitive grasp of the basic notions of topology will often suffice to understand the material treated in this book, we report here some basic definitions and well-known facts in order to make these notes self-contained. The main references for this sections are the books of Abraham et al. [1] and Boothby [11].

D.1.1 Basic topology

Let M be a set. A *topology* on M is the collection of its *open* subsets, which are defined as follows:

1. M and $\emptyset$ are open
2. $U_i \subset M$ are open $\forall i \implies \cup_i U_i$ is open
3. $U_i \subset M$ are open $\implies \cap_{i=1}^{k} U_i$ is open for all *finite* k.

A set with a topology is called a *topological set.* A subset $U \subset M$ is a *neighborhood* of a point $p \in M$ if it is open and it contains p. A set M is *Hausdorff* if any two points have disjoint neighborhoods: $\forall\, p, q \in M,\ \exists\, U, V \subset M$open sets $\mid p \in U, q \in V, U \cap V = \emptyset$. A collection of open subsets $\{U_i\}$ of M is a *basis* if

1. $M = \cup_i U_i$ and
2. $U_i \cap U_j = \cup_{k \in K} U_k$ for a set of indices K

i.e. any non-empty intersection of basis elements can be written as the union of some basis elements. A function $f : M \to N$ between two topological spaces is *continuous* if the pre-image of an open set is open: $V \subset N$ open $\implies f^{-1}(V) \subset M$ open. A continuous function whose inverse is also continuous is called an *homeomorphism.* Two sets are *homeomorphic* if there exists an homeomorphism that maps one into the other.

At this point we are ready to give a formal definition of a topological manifold: M is a *topological manifold* of dimension m if

1. M is Hausdorff

2. M has a countable basis

3. M is locally homeomorphic to $\mathbb{R}^m$.

The third condition states that – around each point – we can find a homeomorphism that "flattens out" the manifold so that it looks like the usual $\mathbb{R}^m$ [figure here]. The pair composed by an open subset $U \subset M$ and a homeomorphism $\phi : U \longrightarrow \mathbb{R}^m; p \mapsto x$ is called a *local coordinate chart*, and $x = \phi(p)$ is called the local coordinate of p. Naturally the local coordinate of a point is not uniquely defined, for we can find two different neighborhoods U and V, and two corresponding coordinate charts (U, ϕ) and (V, ψ), such that $x = \phi(p) \neq \psi(p) = y$. The map $\phi \circ \psi^{-1} : U \cap V \to U \cap V; y \mapsto x$ is a homeomorphism, which is called a *coordinate transformation.* If such a coordinate transformation (which is a function on $\mathbb{R}^m$) is a C^∞ (or smooth) function, i.e. it is infinitely differentiable, then the two coordinate charts are said to be C^∞ *compatible.* A collection of charts that cover the whole manifold M is called an *atlas.* If an atlas is composed by pairwise C^∞ compatible charts, it is called a C^∞ atlas. A topological manifold is called a *smooth manifold* (or differentiable manifold, or C^∞ manifold) if it is equipped with a C^∞ atlas. A smooth homeomorphism, with smooth inverse, is called a *diffeomorphism.*

For example, the sphere is a two-dimensional smooth manifold. To construct a local coordinate chart, imagine to cut the sphere with a plane through the equator, and to draw a line from the north pole, through each point, until it intersects the plane [figure here]. By doing so we can establish a correspondence between points on the sphere and points on the Euclidean plane. It is easy to see that such a correspondence is a smooth function. The point of intersection is the local coordinate of the point on the sphere. Note, however, that we cannot find the local correspondent of the north pole, using this chart. We may indeed repeat the construction from the south pole. In this case the north pole does have a well-defined local coordinate, but the south pole does not. The two local coordinate charts considered cover all points of the sphere, and it can be shown that they are C^∞ compatible. It can also be shown that it is not possible to find a single chart that covers all the sphere, and therefore it is only *locally* diffeomorphic to $\mathbb{R}^2$.

Consider a function F mapping a manifold M into another manifold N, and let (U, ϕ) and (V, ψ) be coordinate charts on M about the point p and on N about the point $q = F(p)$ respectively. The composition of the functions ϕ^{-1}, F and ψ maps the local coordinates of p to the local coordinates of q. Such a function is typically called the *local coordinate correspondent* of the function F:

$$\tilde{F} : \mathbb{R}^m \to \mathbb{R}^n;\ x \mapsto y \tag{D.1}$$

where $x = \phi(p), y = \psi(q), q = F(p)$, and therefore

$$y = \psi \circ F \circ \phi^{-1}(x) \doteq \tilde{F}(x). \tag{D.2}$$

The function F is said to be *smooth* when its local coordinate correspondent $\tilde{F}$ is smooth. We will often omit the tilde and use the same symbol F to denote a function between two manifolds, and its correspondent in local coordinates.

D.1.2 Lie groups

A differentiable manifold which possesses a smooth group structure is called a *Lie group.*

We recall that a set G is a *group* if it closed with respect to an operation, call it $\cdot$

$$\begin{aligned} \cdot : G \times G &\longrightarrow G \\ (p, q) &\mapsto p \cdot q \end{aligned} \tag{D.3}$$

which is associative, has a null element and an inverse:

1. $(p \cdot q) \cdot r = p \cdot (q \cdot r)\ \forall\ p, q, r \in G$ (associative)
2. $\exists\ e \in G \mid g \cdot e = g\ \forall g \in G$ (null element)
3. $\forall\ g \in G\ \exists\ g^{-1} \in G \mid g \cdot g^{-1} = g^{-1} \cdot g = e$ (inverse).

For instance, the set of non-singular $n \times n$ matrices is a group under the usual matrix product, and it is also a differentiable manifold, for it is isomorphic to $\mathbb{R}^{n^2}$. Therefore, it is a Lie group, also called $\mathcal{GL}(n)$,

general linear group acting on $\mathbb{R}^n$.

D.1.3 Embeddings

Let $F : M \longrightarrow N$ be a smooth map between two manifolds. F is said to be an *immersion* if the rank of F equals the dimension of M. The *rank* of a function between manifolds can be defined as the rank of the differential of the local coordinate version $\tilde{F}$ – known from calculus – which is easily shown to be coordinate-independent. The map F is said to be an *embedding* if it is an injective immersion. There are some non-trivial issues concerning the topology induced from M by F and how it compares to the topology of N; the reader may consult for instance [41] for more details.

> We will use often the term *embedding space*. In order to illustrate this concept imagine a manifold, for instance a sphere $S^2 = \{x \in \mathbb{R}^3 \mid \|x\| = 1\}$. Consider now the map $I : S^2 \longrightarrow \mathbb{R}^3; x \mapsto x$, which does not change the point x, but simply views it as an element of the ambient space $\mathbb{R}^3$ without paying attention to the constraint $\|x\| = 1$, which defines the sphere. We say that the map I maps points of a manifold into its embedding space. The embedding (or ambient) space is usually chosen so that it is less structured and therefore easier to work with. As a matter of fact, each differentiable manifold can be embedded in $\mathbb{R}^k$ with k big enough (in fact Withney's Embedding theorem [11] regulates how big k must be).

D.1.4 Tangent plane and tangent bundle

In this paragraph we will describe two equivalent characterizations of the tangent space to a manifold. The link between these two definitions will be hinted to in section D.2. The first one resorts to the intuitive notion of the tangent plane to a surface in $\mathbb{R}^3$: in fact, we can define the tangent space to the manifold M at the point p as the linear space of the derivatives of all possible one-parameter curves of M passing through p [figure here]. Consider a curve $c^i : U \subset \mathbb{R} \longrightarrow M; t \mapsto c^i(t)$, where U is a neighborhood of p, such that there exists some t_0 with $c^i(t_0) = p$. Then we can define

$$T_pM \doteq < \dot{c}^1(t_0), \ldots, \dot{c}^k(t_0), \ldots > \tag{D.4}$$

where $\dot{c} \doteq \frac{dc}{dt}$ and $< \ldots >$ denotes the linear span. It is easy to see that the tangent space thus defined is a linear space of the same dimension of the manifold M, and it corresponds to the intuitive generalization of the tangent plane to a surface in $\mathbb{R}^3$. The tangent *space* is often called tangent *plane* regardless the dimension of the manifold.

An alternative definition, which is less intuitive but easier to work with, uses the concept of a derivation. To this end, consider the set of all smooth real-valued functions defined in a neighborhood of the point p:

$$C^\infty(p) \doteq \{h : U \subset M \to \mathbb{R} \text{ smooth} \mid p \in U\} \tag{D.5}$$

and consider all the real-valued linear functionals on $C^\infty(p)$ that satisfy Leibniz rule:

$$\mathcal{D}(p) = \{f_p : C^\infty(p) \longrightarrow \mathbb{R}; h \mapsto f_p h\} \tag{D.6}$$

such that

1. $f_p(\alpha_1 h_1 + \alpha_2 h_2) = \alpha_1 f_p h_1 + \alpha_2 f_p h_2 \; \forall \; \alpha_1, \alpha_1 \in \mathbb{R}, h_1, h_2 \in C^\infty(p)$ (linearity)
2. $f_p(h_1 h_2) = h_1(p) f_p h_2 + h_2(p) f_p h_2 \; \forall \; h_1, h_2 \in C^\infty(p)$ (Leibniz rule).

Such functionals are called *derivations* on M at p. We then define the *tangent space* of a manifold M at a point p, T_pM, as the space of derivations on M at p:

$$T_pM \doteq \mathcal{D}(p). \tag{D.7}$$

A map $F : M \longrightarrow N; p \mapsto q$ between two smooth manifolds determines in a natural way a map between their tangent planes at p and $q = F(p)$:

$$F_{*_p} : T_pM \longrightarrow T_qN \tag{D.8}$$

that maps a tangent vector f_p on M onto a tangent vector g_q on N according to the rule

$$g_q h = F_{*_p}(f_p)h \doteq f_{F^{-1}(q)} F \circ h. \tag{D.9}$$

Such a map is called the *differential* of F [1], or *push-forward* [11]. It is easy to see that the push-forward of the local-coordinate correspondent of the function, $\tilde{F}$, is the Jacobian matrix of partial derivatives

$$\tilde{F}_{*_p} = \left(\frac{\partial \tilde{F}}{\partial x}\right)_{x=\phi(p)}. \tag{D.10}$$

The latter definition of the tangent plane allows us to easily characterize a *basis* for the tangent plane. Define m tangent vectors to $\mathbb{R}^m$ at x as

$$\frac{\partial}{\partial x_i} \; \forall \; i = 1 \ldots m, \tag{D.11}$$

which is easily seen to be a basis for all derivations in $\mathbb{R}^m$, for all tangent vectors f_x to $\mathbb{R}^m$ at x can be written as $f_x \doteq f_{x_1}\frac{\partial}{\partial x_1} + \ldots f_{x_m} + \frac{\partial}{\partial x_m}$, which acts on a function h as its derivative along the direction f_x:

$$f_x h = \frac{\partial h}{\partial x} f_x. \tag{D.12}$$

We can now consider a local coordinate chart (U, ϕ) as a map between the manifolds M and $\mathbb{R}^m$, which induces a push-forward ϕ_{*_p} between T_pM and $T_x\mathbb{R}^m$, where $x = \phi(p)$. We define a basis of T_pM as the push-forward of the basis of $T_x\mathbb{R}^m$:

$$T_pM = < \frac{\partial}{\partial \phi_1}, \ldots, \frac{\partial}{\partial \phi_m} > \tag{D.13}$$

where

$$\frac{\partial}{\partial \phi_i} \doteq \phi_*^{-1}(\frac{\partial}{\partial x_i}) \tag{D.14}$$

which acts on a function $h \in C^\infty(p)$, where $p = \phi^{-1}(x)$, according to

$$\phi_{*_p}^{-1}(\frac{\partial}{\partial x_i})h \doteq (\frac{\partial h \circ \phi^{-1}}{\partial x_i})_{x=\phi(p)} \tag{D.15}$$

which in local coordinates becomes

$$\frac{\partial \tilde{h}}{\partial x} \tag{D.16}$$

where $\tilde{h}$ is the local coordinate correspondent of h.

It is possible to prove that the tangent plane to a manifold is itself a manifold of the same dimension. If

we consider the collection of all possible tangent planes to all points of a manifold, we obtain

$$TM \doteq \cup_{p \in M} T_p M \tag{D.17}$$

which is called the *tangent bundle.* It is possible to show that the tangent bundle to a manifold is a manifold of twice the dimension. The push-forward of a map $F : M \longrightarrow N$, when the point $p \in M$ is not specified, can be interpreted as a map between the tangent bundles of M and N:

$$F_* : TM \longrightarrow TN. \tag{D.18}$$

D.1.5 Vector fields and Lie derivatives

A (smooth) *vector field* is defined simply as a smooth map between a manifold and its tangent bundle:

$$f : M \longrightarrow TM; p \mapsto f_p \in T_p M. \tag{D.19}$$

Since f_p is a vector in the tangent plane to M at p, it acts on functions $h \in C^\infty(p)$, returning a scalar $f_p h \in \mathbb{R}$. We denote with $\chi(M)$ the set of vector fields defined on the manifold M.

The same object, f, could also be interpreted as a map between smooth real-valued functions on M:

$$f : C^\infty(M) \longrightarrow C^\infty(M); h \mapsto fh \tag{D.20}$$

where the function $fh : M \longrightarrow \mathbb{R}$ acts on a point p via $fh(p) \doteq f_p h$. When used this way, fh is called *Lie derivative* of h along f, and indicated by

$$L_f h \doteq fh : M \longrightarrow \mathbb{R}; p \mapsto L_f h(p) = f_p h. \tag{D.21}$$

In local coordinates the Lie derivative $L_f h$ is, not surprisingly, the directional derivative of h along $f : \frac{\partial h}{\partial x} f$. The *Lie bracket* between two vector fields f_1 and f_2 is defined as follows:

$$[\cdot, \cdot] : \chi(M) \times \chi(M) \longrightarrow \chi(M); \ (f_1, f_2) \mapsto L_{f_1} f_2 - L_{f_2} f_1. \tag{D.22}$$

In local coordinates we have $[f_1, f_2] = \frac{\partial f_2}{\partial x} f_1 - \frac{\partial f_1}{\partial x} f_2$.

D.1.6 Duality

V^*, the *dual* of a vector space V, is defined as the space of linear functionals on V. V^* is itself a vector space, and its elements, w, act on the elements v of V giving a value $< w, v > \in \mathbb{R}$. If $\{e_1, \ldots, e_m\}$ is a basis of V, the unique vectors $\{e_1^*, \ldots, e_m^*\}$, defined by $< e_i^*, e_j >= \delta_{ij}$ [1], are a basis of V^*, called the dual basis. Given a map $F : V \longrightarrow W$ between two vector spaces, the *dual map* $F^* : W^* \longrightarrow V^*$ is defined by

$$< F^*(w^*), v >=< w^*, F(v) > \forall v \in V, w^* \in W^*. \tag{D.23}$$

For instance, the *cotangent space* to a manifold M at a point p, T_p^*M is the dual of the tangent space; the *tangent covectors* are the dual of tangent vectors, and the *covector fields* are dual of vector fields.

A function $\lambda : M \longrightarrow$ generates a natural covector field, $d\lambda$, which is called the *gradient* (or differential) of λ, such that

$$\lambda_*(f) \doteq < d\lambda, f > \frac{d}{dt}. \tag{D.24}$$

The differential is expressed in local coordinates as the row vector of partial derivatives of λ with respect to $x_1, \ldots, x_m$:

$$d\lambda(x) = [\frac{\partial \lambda}{\partial x_1}, \ldots, \frac{\partial \lambda}{\partial x_m}]. \tag{D.25}$$

D.2 Differential equations, local flows and one-parameter group actions on a manifold

The notion of a vector field allows us to introduce the concept of a differential equation on a manifold. In fact, consider a curve on a manifold M described by

$$p : (t_1, t_2) \longrightarrow M; t \mapsto p(t). \tag{D.26}$$

[1] $\delta_{ij} = \{1 \text{ if } i = j, 0 \text{ otherwise}\}$ is the Kronecker delta.

The differential of this map at t, p_{*_t} maps the vector $(\frac{d}{dt})_t$ onto a vector in the tangent plane to M at $p(t) : p_{*_t}(\frac{d}{dt}) \doteq \dot{p}(t)$. Therefore, we have

$$p_* : T\mathbb{R} \longrightarrow TM; (\frac{d}{dt}) \mapsto \dot{p}(t) \doteq p_*(\frac{d}{dt}). \tag{D.27}$$

Now assume we are given a vector field $f : M \longrightarrow TM$, the curve $\{p(t)\}$ described above is an *integral curve* of f if the following is satisfied locally around t:

$$\dot{p}(t) = f_{p(t)}. \tag{D.28}$$

We will often write $\dot{p} = f(p)$ or even, in local coordinates, $\dot{x} = f(x)$.

We will now see that there is a close relationship between vector fields and one-parameter group actions on a manifold, which is essentially governed by the fundamental theorem of ordinary differential equations.

> Such a relationship glues together the two definitions of the tangent plane that we have given in section D.1.4. The reader interested in a more detailed and rigorous treatment of this issue can see [11].

D.2.1 Group actions and infinitesimal generators

We define the (smooth) *action of a group G on a set M* as follows as follows:

$$\phi : G \times M \longrightarrow M; \ (t, p) \mapsto \phi(t, p) \tag{D.29}$$

such that

1. $\phi(e, p) = p \ \forall \ p \in M$, where e is the null element of the group G,
2. $\phi(t_1, \phi(t_2, p)) = \phi(t_1 \cdot t_2, p) \ \forall \ t_1, t_2 \in G, p \in M$ where $\cdot$ is the group operation (semi-group property),
3. ϕ is a smooth map.

We will often use the notation $\phi_t(p)$ or $\phi_p(t)$ in place of $\phi(t, p)$ when emphasizing the role of the group element t or the point the group is acting on, p. ϕ are called *trajectories*; when the action takes the trajectories out

of M, $\phi : G \times M \to N \subset M$, they can be visualized as *fibers*, organized in a *bundle* [figure here].

Consider now $\mathbb{R}$ as a one-dimensional compact group acting on the manifold M through $\phi(t,p), t \in \mathbb{R}, p \in M$ as above. The group operation is the usual addition: $t_1 \cdot t_2 = t_1 + t_2$. Then we can establish a *local* correspondence between a one-parameter group action $\phi(t,p)$ and a vector field $f \in \chi(M)$. In fact, given a group action $\phi_p(t)$, we can define a vector field f, which is called the *infinitesimal generator*, such that

$$f_p h \doteq \lim_{\Delta t \to 0} \frac{h(\phi_p(t+\Delta t)) - h(p)}{\Delta t}. \tag{D.30}$$

Vice-versa, given a vector field $f \in \chi(M)$, we can define a *local* one-parameter group action, which is called the *flow* of the vector field, through the fundamental theorem of ordinary differential equations. The flow represents an integral curve of the vector field, and therefore it satisfies

$$\dot{\phi}_p(t) = f_{\phi_p(t)}. \tag{D.31}$$

If we add the condition that $\phi_p(0) = p$, then the correspondence between the vector field and its flow becomes (locally) one-to-one. Equation (D.31) represents a differential equation on the manifold M. We will also write

$$\dot{p} = f(p) \tag{D.32}$$

letting $p(t)$ play the role of the flow (trajectory) $\phi_p(t)$ or, in local coordinates,

$$\dot{x} = f(x). \tag{D.33}$$

Rigorous proofs of the above statements are beyond the scope of this book, and can be found in standard texts on differential geometry, such as [1, 11].

D.2.2 Action on Lie groups; exponential coordinates

When a one-dimensional compact group acts on a manifold, we can establish a local correspondence between vector fields and group actions, interpreted as a flow of a system of differential equations. If the manifold where the one-dimensional group is acting is itself a Lie group, then it is possible to unravel the structure of

the flow even further.

Consider a matrix Lie group with the product operation (either the left product $\mathcal{L}_{p_1}p_2 = p_1 \cdot p_2 \doteq p_1p_2 \ \forall \ p_1, p_2 \in G$, or the right product $\mathcal{R}_{p_1}p_2 = p_1 \cdot p_2 = p_2p_1$). Consider the class of vector fields $\mathcal{L}(G)$ (or $\mathcal{R}(G)$) that are invariant under left (right) multiplication: $\mathcal{L}_* f = f \ \forall \ f \in \chi$. Then, given a tangent vector f at any point p, it is possible to push it forward to the tangent plane to the origin by left (right) multiplication with the inverse p^{-1}

$$f_p \in T_pG \mapsto \mathcal{L}_{*p^{-1}} f_p \in T_eG \tag{D.34}$$

where e is the identity vector (origin) of the group G. Now, each vector f_e tangent to the origin e of G defines a unique one-parameter subgroup of G via

$$f_e \in T_eG \mapsto e^{f_e t} \in G. \tag{D.35}$$

It is possible to prove that all an only the (compact) one-parameter subgroups of a Lie group are exponentials of the above form [95].

Therefore, in the case of a Lie group, it is possible to establish a local one-to-one correspondence between left (right) invariant vector fields, the tangent plane to the group at the origin, and the exponential group action. The tangent plane to the origin of a Lie group is a Lie algebra. A vector space V is a *Lie algebra* if we can define an operation $[\cdot, \cdot] : V \times V \longrightarrow V$, called *bracket*, that satisfies the following three conditions:

1. $[\alpha_1 v_1 + \alpha_2 v_2, w] = \alpha_1 [v_1, w] + \alpha_2 [v_2, w] \ \forall \ \alpha_1, \alpha_2 \in \mathbb{R}, v_1, v_2, w \in V$ (linearity)
2. $[v_1, v_2] = -[v_2, v_1] \ \forall \ v_1, v_2 \in V$ (skew-symmetry)
3. $[u, [v, w]] + [v, [w, u]] + [w, [u, v]] = 0 \ \forall \ u, v, w \in V$ (Jacobi identity).

The set of vector fields on a manifold $\chi(M)$ is a Lie algebra with the bracket $[f, g]$ defined in eq. (D.22) such that it acts on a smooth function h via

$$[f, g]_p h \doteq L_f L_g h(p) - L_g L_f h(p). \tag{D.36}$$

It is possible to characterize the Lie bracket in terms of the flow of the vector fields involved. Call $\phi_t^f(p)$ the

flow along f starting at p. Then for each $p \in M$ we have [11]

$$\lim_{t \to 0} \frac{\phi^f_{*-t} g \circ \phi^f_t(p) - g_p}{t} = [f, g](p). \tag{D.37}$$

D.3 Distributions and Frobenius theorem

A *distribution* is a (smooth) assignment

$$\Delta : M \longrightarrow TM \tag{D.38}$$

such that there exist d vector fields $f_1, \ldots, f_d$ with $p \mapsto \Delta(p) = \text{span}\{f_{1p}, \ldots, f_{dp}\}$, where the span is intended with coefficients on $C^\infty(M)$. Note that, for each p fixed, $\Delta(p)$ is a vector space, so that it is possible to extend notions such as the sum, intersection and dimension of a distribution directly from linear spaces. In particular, we say that a distribution is *non-singular* on M if it has constant dimension r. In the local coordinates, $\text{rank}(\Delta(x)) = r \ \forall \ x \in \tilde{U}$, where U is a neighborhood of a point p. It is possible to define a *basis* of a distribution around a regular (non-singular) point, i.e. a family of vector fields $\{f_1, \ldots, f_d\}$ such that

1. $f_1 \ldots f_d$ are linearly independent for each $x \in U$,
2. $f_1 \ldots f_d$ span the distribution,
3. $\forall \ \tau \in \Delta, \tau(x) = \sum_{i=1}^d c_i(x) f_i(x) \ \forall x \in U$, where c_i are smooth functions on U.

A distribution Δ is said to be *involutive* if it is closed under the bracket operation: $\tau_1, \tau_s \in \Delta \Longrightarrow [\tau_1, \tau_2] \in \Delta$. Note that it is possible to check the involutivity of the distribution by verifying that the brackets of the basis elements are elements of the distribution.

Similarly, a *co-distribution* is a (smooth) assignment of elements of the co-tangent bundle:

$$\Omega : M \longrightarrow T^*M \tag{D.39}$$

spanned by covector fields. Given a distribution Δ, if there exists a co-distribution, Ω, that annihilates it, in the sense that $< \omega^*, v >= 0 \ \forall \ \omega^* \in \Omega, \ \forall \ v \in \Delta$, we say that Ω is the *annihilator* of Δ, and we indicate it by $\Omega = \Delta^\perp$.

Consider now a non-singular distribution Δ of dimension d, spanned by $\{f_1, \ldots, f_d\}$ around a point p, with local coordinate x. Let $\Omega = \Delta^\perp$ be a non-singular co-distribution of dimension $m-d$, which is spanned, locally around x, by $\{\omega_1, \ldots, \omega_{m-d}\}$, such that $< \omega_j, f_i >= 0 \; \forall i = 1, \ldots, d; \forall j = 1, \ldots, m-d$. In local coordinates we can collect all the vector fields f_i into a matrix F, and the above condition becomes

$$\omega_j F(x) = 0 \; \forall \; j = 1, \ldots, m-d \tag{D.40}$$

where $F = [f_1 \ldots f_d]$. Now imagine to seek, among all covector fields ω_j, those which are *exact differentials* of some functions λ_j, i.e.

$$\omega_j = \frac{\partial \lambda_j}{\partial x}. \tag{D.41}$$

The above problem corresponds to finding $m-d$ independent solutions to the first-order partial differential equation (PDE)

$$\frac{\partial \lambda_j}{\partial x} F(x) = 0 \; \forall \; j = 1 \ldots m-d. \tag{D.42}$$

When we can solve this problem, we say that the distribution Δ is *completely integrable*, i.e. when there exist, locally around x, $m-d$ functions $\lambda_1, \ldots, \lambda_{m-d}$ such that $\Delta^\perp = \text{span}\{d\lambda_1, \ldots, d\lambda_{n-d}\}$. Frobenius theorem [11] states that integrability, such a strong condition on a distribution, is equivalent to involutivity, which can be verified by simple computations (see [11]).

D.3.1 Flat distributions

Let Δ be a non-singular, d-dimensional involutive distribution spanned by $\{f_1, \ldots, f_d\}$, and call $\{\lambda_1, \ldots, \lambda_d\}$ the functions $\lambda_i : M \longrightarrow \mathbb{R}$ whose differentials satisfy the PDE $\frac{\partial \lambda_i}{\partial x} F(x) = 0 \; \forall \; i = 1 \ldots m-d$ (such functions exist according to Frobenius theorem) locally around some $x_0 \in M$. The one-forms $\{d\lambda_1, \ldots, d\lambda_{m-d}\}$ are independent and span the annihilating codistribution $\Delta^\perp$. Let us change coordinates using the $m-d$

functions λ and d other functions ϕ_i in order to complete the basis,

$$\Phi \doteq \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_d \\ \lambda_1 \\ \dots \\ \lambda_{m-d} \end{bmatrix} : U \longrightarrow \mathbb{R}^m. \tag{D.43}$$

It is easy to show that, in the new coordinates, all vectors $\tau \in \Delta$ are characterized by having the last $m-d$ components identically zero. In fact, consider Φ as a function, $\Phi : M \subset \mathbb{R}^m \longrightarrow M \subset \mathbb{R}^m$, that maps a point p into $\Phi(p)$. The action induced by Φ on a vector field τ is the push-forward

$$\tau(p) \mapsto \bar{\tau}(q) = \Phi_* \tau(q)|_{q \ \Phi^{-1}(p)} \tag{D.44}$$

which can be computed in local coordinates as

$$\bar{\tau}(z) = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} \tau(\phi^{-1}(z)) \\ \vdots \\ \frac{\partial \lambda_1}{\partial x} \tau(\phi^{-1}(z)) \\ \vdots \end{bmatrix} \tag{D.45}$$

where z is the local coordinate of q and $x = \phi^{-1}(z)$ is the local coordinate of p. Now, since $d\lambda_i$ span the codistribution that annihilates Δ, and $\tau \in \Delta$, we have necessarily $\frac{\partial \lambda_i}{\partial x} \tau(\phi^{-1}(z)) = 0 \ \forall \ i = 1 \dots m-d$. Hence,

there is a set of coordinates where the last $m-d$ components of the vector fields of the distribution are zero:

$$\bar{\tau} = \begin{bmatrix} \bar{\tau}_1 \\ \vdots \\ \bar{\tau}_d \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{D.46}$$

Such coordinates are called *flat coordinates*, or *flow-box coordinates*, and the distribution is said to be *flat* in these coordinates.

A subset of the manifold M is called a d-dimensional *slice* S at the point p if there exists a neighborhood U of p such that, for all $q \in U$ we have

$$\lambda_{d+1}(q) = \lambda_{d+1}(p), \ldots, \lambda_m(q) = \lambda_m(p). \tag{D.47}$$

Therefore, an involutive distribution naturally identify a slice of a manifold; in fact, the tangent plane to the slice is exactly the distribution:

$$T_q S = \Delta(q) \subset T_q M \ \forall \ q \in S. \tag{D.48}$$

D.3.2 Invariant distributions

We say that a distribution Δ is *invariant* under the vector field f if it is closed under the bracket operation with f:

$$\tau \in \Delta \implies [\tau, f] \in \Delta \ \forall \ \tau \in \Delta. \tag{D.49}$$

It is customary to write $[f, \Delta] \subset \Delta$. We will now see that an involutive distribution which is f-invariant allows writing f in a special "triangular" form.

Let Δ be a d-dimensional involutive distribution which is f-invariant, and let us use the same change of coordinates (D.43) described in the previous subsection. Then the generic vector field $\tau \in \Delta$ has the form of equation (D.46), with the last $m-d$ coordinates equal to zero. Let us divide f into the first d components f^1,

and the last $m-d$, f^2, and similarly for the point p and the vector field τ. From the previous considerations we have $\tau^2 = 0$ for the generic vector field in the distribution Δ. Now, since all brackets of the elements of Δ with f must also be elements of Δ, they must be of the form

$$\left[\begin{bmatrix} f^1(p^1,p^2) \\ f^2(p^1,p^2) \end{bmatrix}, \begin{bmatrix} \tau^1 \\ 0 \end{bmatrix}\right] = \begin{bmatrix} * \\ 0 \end{bmatrix} \tag{D.50}$$

where the asterisk $*$ indicates some vector field. Writing the above bracket in local coordinates we get

$$\begin{bmatrix} \frac{\partial f^1}{\partial x^1} & \frac{\partial f^1}{\partial x^2} \\ \frac{\partial f^2}{\partial x^1} & \frac{\partial f^2}{\partial x^2} \end{bmatrix} \begin{bmatrix} \tau^1 \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{\partial \tau^1}{\partial x^1} & \frac{\partial \tau^1}{\partial x^2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f^1 \\ f^2 \end{bmatrix} = \begin{bmatrix} * \\ \frac{\partial f^2}{\partial x^1}\tau^1 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}. \tag{D.51}$$

Therefore, since $\frac{\partial f^2}{\partial x^1} = 0$, the portion of the vector field f^2 depends only upon x^2. If the vector field f is associated to a differential equation, then its general form is

$$\begin{cases} \dot{x}^1 = f^1(x^1,x^2) \\ \dot{x}^2 = f^2(x^2). \end{cases} \tag{D.52}$$

Therefore, we have seen that an involutive distribution generates slices of the manifold M. If the distribution is invariant under a vector field f, then the flow of the vector field f carries slices into slices:

$$S = \{z \in M \mid z = \phi^{f^2}(p); p \in S\}. \tag{D.53}$$

The same reasoning can be carried out for involutive co-distributions, as we will outline in the following example. We remind that a co-distribution Ω is invariant under a vector field f if $L_f\omega \in \Omega \; \forall \; \omega \in \Omega$.

> Consider a vector field $\dot{x} = f(x)$, which describes the evolution of some state $x \in \mathbb{R}^m$. Assume that we can measure the "output" of such evolution, which is in the form of some function of the state $y = h(x) \in \mathbb{R}^k$. Now, suppose that there exists a co-distribution Ω that is involutive, d-dimensional, f-invariant and that it contains the one-form dh. If we decompose h into its first $m-d$ components, h^1, and the last d components, h^2 then, from the discussion of the previous paragraphs, we know that $dh^1 = 0$ and, therefore, locally in the flow-box coordinates, $h = h(x^2)$.

At the same time, because Ω is f-invariant, f is such that $\frac{\partial f^2}{\partial x^1} = 0$. Therefore, the original system is transformed into

$$\begin{cases} \dot{x}^1 = f^1(x^1, x^2) \\ \dot{x}^2 = f^2(x^2) \\ y = h(x^2). \end{cases} \tag{D.54}$$

What we can measure is

$$y(t)_{t\in[t_0,t_f]} = h(\phi^{f^2}_{x_0}(t))_{t\in[t_0,t_f]} \tag{D.55}$$

which is a function of the evolution of x^2 alone. Because x^2 evolves independently of x^1, we will never be able to assess the evolution of x^1 by measuring y. Therefore, the points x that are on the form $\begin{bmatrix} z^1 \\ x^2 \end{bmatrix}$ produce the same output y as the points $\begin{bmatrix} w^1 \\ x^2 \end{bmatrix}$ for $w^1 \neq z^1$. In other words they are *indistinguishable* from the output.

All the points that are indistinguishable from $x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$ are therefore of the form $\{ \begin{bmatrix} w^1 \\ x^2 \end{bmatrix} \mid w^1 \in \mathbb{R}^{m-d} \}$, which identifies a *slice* of M. However, it is not clear that *only* the points of that form are indistinguishable. We can guarantee uniqueness only if we assume that Ω is the *smallest* (in the sense of inclusion) involutive co-distribution that contains the forms dh. Such a co-distribution can be constructed incrementally by taking the one forms dh, then taking all possible Lie derivatives along f until the co-distribution becomes involutive:

$$\begin{aligned} \Omega_0 &= \{dh\} \\ \Omega_1 &= \{\Omega_0, L_f\Omega_0\} \\ \vdots &= \vdots \\ \Omega_r &= \{\Omega_{r-1}, L_f\Omega_{r-1}\} \end{aligned} \tag{D.56}$$

where $L_f\Omega$ denotes all Lie derivatives of the elements of the co-distribution Ω along the vector field f. If there exists an r such that $\Omega_r = \Omega_{r+1}$, then Ω_r is the smallest involutive co-distribution that contains the forms dh. Note in fact that the sequence of co-distributions is increasing (in the sense of inclusion), and generates an involutive co-distribution. The problems may arise if

the co-distribution is singular at a point, in which case we may only guarantee that the sequence above generates the smallest involutive co-distribution which contains *dh in an open and dense subset* of a neighborhood of the point.

D.4 Fundamentals of the Euclidean group

Consider a rigid object in the three-dimensional space. Euclidean geometry is concerned with the transformations g of the three-dimensional space that preserve the distance between any two points $p_i; i = 1, 2$ of the space, and the cross product between any two vectors $q_i; i = 1, 2$:

$$\begin{aligned} d(p_1, p_2) &= d(g(p_1), g(p_2)) \quad \forall\, p_1, p_2 \in \mathbb{R}^3 \\ g_*(q_1 \wedge q_2) &= g_*(q_1) \wedge g_*(q_2) \quad \forall\, q_1, q_2 \in T\mathbb{R}^3 \sim \mathbb{R}^3 \end{aligned}$$

where g_* is the transformation induced on vectors: $q \doteq p_2 - p_1 \Rightarrow g_*(q) \doteq g(p_2) - g(p_1)$, and $T\mathbb{R}^3$ is the tangent space to $\mathbb{R}^3$. Such transformations are called *congruencies*, *similarities* or *rigid motions.*

If we take $\mathbb{R}^3$ as a model of the Euclidean space, with the Hilbert structure relative to the inner product $< \cdot, \cdot >: \mathbb{R}^3 \times \mathbb{R}^3 \longrightarrow \mathbb{R}; (p_1, p_2) \mapsto < p_1, p_2 > \doteq p_1^T p_2$, and we represent the points p_i in coordinates $\mathbf{X}_i \doteq [X_i\ Y_i\ Z_i]^T$ relative to some orthonormal reference frame, then it is easy to show that the congruencies are all and only the transformations of the space of the form

$$g : \mathbb{R}^3 \longrightarrow \mathbb{R}^3; \mathbf{X} \mapsto R\mathbf{X} + T \tag{D.57}$$

where $T \in \mathbb{R}^3$ and R is an orthonormal matrix with unit determinant (to rule out reflections), i.e. such that $RR^T = R^T R = I$, and I is the identity matrix. Intuitively, imagine a rigid object in the three dimensional space, and consider a point in the object as the *origin* of an orthonormal coordinate system fixed in the object, such that the coordinate vectors satisfy $< e_i, e_j >= \delta_{ij}\ \forall\, i, j = 1 \ldots 3$. The rigid motion of the object can be described, relative to a *world* reference frame, as a translation of the origin and a rotation of the *object* reference. No deformation that destroys the orthonormality is allowed, and also reflections are excluded, since we require that the determinant of R is positive. The transformation (D.57) can also be interpreted as

a change of coordinates in $\Re^3$ that preserves distances between points and angles between vectors of $\mathbb{R}^3$.

The set of orthonormal matrices with positive determinant is a group under the usual matrix multiplication, called $SO(3)$ (special orthogonal group in $\mathbb{R}^3$). The identity matrix is the null elment, and the inverse matrix exists, for the determinant is non-zero. It is possible to prove that it is also a smooth manifold, using the pre-image theorem [41] for the function $o : \mathbb{R}^3 \longrightarrow \mathbb{R}^3; R \mapsto R^T R$. The pre-image theorem shows that $SO(3) \doteq o^{-1}(I)$, where I is the identity matrix, is a differentiable manifold of dimension 3. Therefore, $SO(3)$ is a Lie group.

A rigid motion is described by a pair (T, R), where $T \in \mathbb{R}^3$ and $R \in SO(3)$. It is immediate to prove that the space $SE(3) \doteq \mathbb{R}^3 \times SO(3)$ is also a Lie group, called *special Euclidean group*, and that it has dimension 6 (three dimensions pertain to rotation, and three to translation). We denote with $g = (T, R)$ a rigid motion, intended as an element of the Euclidean group $SE(3)$, which acts on $\mathbb{R}^3$ as follows:

$$\phi : SE(3) \times \mathbb{R}^3 \longrightarrow \mathbb{R}^3; (g, p) \mapsto gp; \text{ in coordinates } R\mathbf{X} + T. \tag{D.58}$$

If we *embed* $SE(3)$ in the matrix group $\mathcal{GL}(4)$, we can represent the group action of $SE(3)$ on $\mathbb{R}^3$ as a matrix multiplication:

$$\bar{\mathbf{X}} \mapsto G\bar{\mathbf{X}} \tag{D.59}$$

where

$$G \doteq \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}, \text{ and } \bar{\mathbf{X}} \doteq \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \tag{D.60}$$

are called *homogeneous coordinates.*

Let us now characterize the tangent vectors to $SO(3)$. It is easy to show that the tangent plane to the origin of $SO(3), T_I SO(3)$ is exactly the set of skew-symmetric 3×3 matrices [95]:

$$T_I SO(3) \doteq so(3) = \{S \in \mathbb{R}^{3\times 3} \mid S^T = -S\}. \tag{D.61}$$

The reader can verify that the matrix exponential of a skew-symmetric matrix S is a rotation matrix:

$$R = e^{St} \in SO(3) \; \forall \; S \in so(3); t \in \mathbb{R}. \tag{D.62}$$

We can now ask ourselves how the tangent vectors to a generic point $R \in SO(3)$ are made. They are nothing else than tangent vectors to the origin, i.e. elements of $so(3)$, pushed-forward to the point R via, for instance, right multiplications: SR.

In fact, given an element g_1 of a matrix Lie group G, and a tangent vector to G at g_1: $\dot{g}_1 \in T_{g_1}G$, the push-forward of a left-multiplication by g_2, $L_{g_2}g_1$ is given by

$$L_{g_2*}\dot{g}_1 = g_2\dot{g}_1 \tag{D.63}$$

for, if we call $\dot{g}_1 \doteq \phi_{*t}(g_1)\frac{d}{dt}$, we have $L_{g_2*}\dot{g}_1 h = \dot{g}_1(h \circ L_{g_2}(g_1) = \phi_{*t}(g_1)\frac{d}{dt}(h \circ L_{g_2}(g_1)) = \frac{d}{dt}(h \circ g_2 g_1) = (g_2 g_1)_{*t}\frac{d}{dt}h = g_2\phi_{*t}\frac{d}{dt}h = g_2\dot{g}_1 h$, and similarly for a right multiplication. Therefore, we can push forward tangent vectors by either left of right multiplications. For instance, given a tangent vector $\dot{g}$ to G at g, we can obtain a tangent vector V^s at the origin e simply by left multiplication

$$V^s = g^{-1}\dot{g} \in T_eG \tag{D.64}$$

or by right multiplication

$$V^b = \dot{g}g^{-1} \in T_eG. \tag{D.65}$$

In the case of a rigid motion, $G = SE(3)$, V^s and V^b are a short way of coding the velocity of the rigid body, and are called *spatial velocity* and *body velocity* respectively. In fact, while $\dot{g} \in T_gSE(3)$ is embedded in $T_G\mathcal{GL}(4)$, which is represented by 12 numbers, V^s or $V^b \in T_eSE(3)$ are represented, as we will see shortly, using 6 numbers.

From what we have seen, the tangent plane to the rotation group $SO(3)$ at a point R is therefore of the form

$$T_RSO(3) = \{SR \mid S \in so(3)\} \tag{D.66}$$

and the tangent bundle is obviously

$$TSO(3) = \{SR \mid S \in so(3), R \in SO(3)\}. \tag{D.67}$$

We will encounter often the space $TSO(3)$, for it plays a crucial role in vision problems.

The space $so(3)$, the Lie algebra corresponding to $SO(3)$, is composed by vectors of the form

$$s\wedge = \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix} \mid s = [s_1, s_2, s_3]^T \in \mathbb{R}^3. \tag{D.68}$$

Therefore, there is a one-to-one global correspondence between skew-symmetric 3×3 matrices and three-dimensional vectors. In this sense $so(3)$ is *isomorphic* to $\mathbb{R}^3$. The notation $s\wedge$ comes from the fact that the cross product between two vectors s_1 and s_2 in $\mathbb{R}^3$, $s_1 \wedge s_2$, can be written as the product of the skew-symmetric matrix $s_1\wedge$ times the vector s_2 : $(s_1\wedge)s_2$.

The isomorphism between $\mathbb{R}^3$ and $so(3)$, together with the exponential map, provides a local coordinatization of $SO(3)$ as follows. Given a three-dimensional vector $s = [s_1, s_2, s_3]^T$, we can construct a skew-symmetric matrix $s\wedge$, and then take the exponential in order to obtain a unique rotation matrix

$$R = e^{s\wedge}. \tag{D.69}$$

It is possible to prove, using Rodrigues' formulae [81], that the converse is also true (although only locally), i.e. given a rotation matrix R, it is possible to take its *logarithm* in order to obtain a skew-symmetric matrix S, and then extract s such that $S = s\wedge$. We have therefore established a local diffeomorphism between $SO(3)$ and $\mathbb{R}^3$, which confirms that $SO(3)$ is a differentiable manifold of dimension 3. This local coordinatization, which is often called *canonical* or *exponential* coordinatization, puts each point $R \in SO(3)$ in correspondence with a tangent vector to the origin $S \in so(3)$, defined such that $R = e^S$.

In a similar way it is possible to establish a system of local coordinates for $SE(3)$, by putting a pair $(T, R) \in SE(3)$ in correspondence with a pair (V, S), where $V \in \mathbb{R}^3$ and $S \in so(3)$ via exponential coordinates. Such coordinates are called *twists* in the robotics literature [81], and may be represented in so-called "Plücker coordinates" as $v\wedge \doteq \dot{g}g^{-1} = \begin{bmatrix} \Omega\wedge & V \\ 0 & 0 \end{bmatrix}$, where $V \in \mathbb{R}^3$ and $\Omega\wedge \in so(3)$. We will use the same symbol for an element of $se(3)$ and its Plücker coordinates. The reader interested in a complete treatment of the concepts sketched here may consult for instance [11, 59, 68, 81, 95]. An explicit expression for the

exponential map on $SE(3)$ is given by

$$\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} = \exp\left(\begin{bmatrix} \Omega\wedge & V \\ 0 & 0 \end{bmatrix}\right)$$

where

$$R \doteq e^{(\Omega\wedge)} \tag{D.70}$$

$$T \doteq \mathcal{T}(\Omega)V \tag{D.71}$$

$$\mathcal{T}(\Omega) \doteq \frac{1}{\|\Omega\|}\left[\left(I - e^{(\Omega\wedge)}\right)(\Omega\wedge) + \Omega\Omega^T\right]. \tag{D.72}$$

The exponential map may be inverted locally for computing V and Ω from R and T, since the matrix $\mathcal{T}(\Omega)$ is invertible when $\|\Omega\| \in (0\ ,\ \pi)$. In the case $\|\Omega\| = 0$, the exponential map is defined simply by

$$R \doteq I \tag{D.73}$$

$$T \doteq V. \tag{D.74}$$

The exponential map, together with the isomorphism of $so(3)$ with $\mathbb{R}^3$, gives a local coordinate parametrization of $SE(3)$, which in the robotics literature is called the "canonical" (exponential) representation. If we consider the composite action of time on the Euclidean space through $SE(3)$, we can motivate the characterization of $v\wedge = \dot{g}g^{-1}$ as "velocity". Consider a point p which has moved between t_0 and t according to some motion: $p(t) = g(t)p(t_0)$. Then we have

$$\dot{p}(t) = \dot{g}(t)p(t_0) = \dot{g}(t)g^{-1}(t)g(t)p(t_0) = v(t) \wedge p(t)$$

and, in coordinates,

$$\dot{\mathbf{X}}(t) = \Omega(t) \wedge \mathbf{X}(t) + V(t), \tag{D.75}$$

where V and Ω represent the translational and rotational velocities of the viewer's moving frame [81].

Note, as a simple observation, that there is a diffeomorphism between $SE(3)$, the Euclidean group

of rigid motions, and $TSO(3)$, the tangent bundle to the rotation group. In fact, given a rigid motion (T, R), with $T \in \mathbb{R}^3$ and $R \in SO(3)$, we can exploit the isomorphism between $\mathbb{R}^3$ and $so(3)$ to construct $S \doteq T\wedge \in so(3)$, and then $SR \in TSO(3)$ is uniquely determined. Vice-versa, *if we are able to separate the factors S and R from the product $SR \in TSO(3)$*, we can then identify the unique T such that $T\wedge = S$, and determine uniquely the element $(T, R) \in SE(3)$ corresponding to $T \wedge R \in TSO(3)$.

This simple observation will turn out to be of primary importance for the vision problem. In fact, instead of representing a rigid motion using the exponential coordinates $(V, \Omega) \in \mathbb{R}^6$ such that

$$e^{\begin{bmatrix} \Omega\wedge & V \\ 0 & 0 \end{bmatrix}} \subset \mathbb{R}^{4\times 4} \tag{D.76}$$

belongs to the Euclidean group $SE(3)$, we represent a rigid motion using (T, Ω) such that

$$T \wedge e^{\Omega\wedge} \subset \mathbb{R}^{3\times 3} \tag{D.77}$$

belongs to the tangent bundle of the rotation group $TSO(3)$. This representation will result in a very natural and well-known constraint on the image of points moving rigidly in a scene. Elements of $TSO(3)$ are called *essential matrices*.

Appendix E The linear Kalman filter

E.1 Least-variance estimators of random vectors

Let $T : \mathbb{R}^n \to \mathbb{R}^m;\ X \mapsto Y$ be a transformation acting between two spaces of random vectors with instances in $\mathbb{R}^m$ and $\mathbb{R}^n$ (the model generating the data). We are interested in building an estimator for the random vector X, given measurements of instances of the random vector Y. An estimator is a function $T^* : \mathbb{R}^m \to \mathbb{R}^n;\ Y \mapsto \hat{X} = T^*(Y)$, which solves an optimization problem of the form

$$\hat{T}^* \doteq \arg\min_{T \in \mathcal{T}} \mathcal{C}(X - T^*(Y))_{\mathcal{T}} \tag{E.1}$$

where $\mathcal{T}$ is a suitable chosen class of functions and $\mathcal{C}(\cdot)_{\mathcal{T}}$ some cost in the $X-$space.

We concentrate on the simplest possible choices, which correspond to *minimum variance affine* estimators:

$$\mathcal{T} \doteq \{A \in \mathbb{R}^{n \times m}; b \in \mathbb{R}^n \mid T^*(Y) = AY + b\} \tag{E.2}$$

$$\mathcal{C}(\cdot)_{\mathcal{T}} \doteq E\|\cdot\|^2 \tag{E.3}$$

where the latter operator takes the expectation of the squared euclidean norm of the random vector Y. Therefore, we seek for

$$(\hat{A}, \hat{b}) \doteq \arg\min_{A,b} E\|X - AY - b\|^2 \tag{E.4}$$

We call $\mu_X \doteq EX$ and $\Sigma_X \doteq EXX^T$, and similarly for Y. First notice that if $\mu_X = \mu_Y = 0$, then $\hat{b} = 0$. Therefore, consider the centered vectors $\bar{X} \doteq X - \mu_X$ and $\bar{Y} \doteq Y - \mu_Y$ and the reduced problem

$$\hat{A} \doteq \arg\min_{A} E\|\bar{X} - A\bar{Y}\|^2. \tag{E.5}$$

Now observe that

$$E\|X - AY - b\|^2 = E\|A\bar{Y} - \bar{X} + (A\mu_X + b - \mu_Y)\|^2 = E\|\bar{X} - A\bar{Y} - b\|^2 + \|A\mu_X + b - \mu_Y\|^2. \quad \text{(E.6)}$$

Hence, if we assume for a moment that we have found $\hat{A}$ that solves the problem (E.5), then trivially

$$\hat{b} = \mu_X - \hat{A}\mu_Y \quad \text{(E.7)}$$

annihilates the second term of eq. (E.6).

Therefore, we will concentrate on the case $\mu_X = \mu_Y = 0$ without loss of generality.

E.1.1 Projections onto the range of a random vector

The set of all random variables Z_i defined on the same probability space, with *zero-mean* $EZ_i = 0$ and *finite variance* $\Sigma_{Z_i} < \infty$ is a *Hilbert space* with the inner-product given by

$$< Z_i, Z_j >_{\mathcal{H}} \doteq \Sigma_{Z_i Z_j} = EZ_i Z_j. \quad \text{(E.8)}$$

In this space the notion of orthogonality corresponds to the notion of *uncorrelatedness*. The components of a random vector Y define a subspace of such Hilbert space:

$$\mathcal{H}(Y) = \text{span} < Y_1, \ldots, Y_m > \quad \text{(E.9)}$$

where the span is intended over the reals. We say that the subspace $\mathcal{H}(Y)$ is *full rank* if $\Sigma_Y = EYY^T > 0$.

The structure of a Hilbert space allows us to make use of the concept of *orthogonal projection* of a random variable onto the span of a random vector:

$$\begin{aligned} \hat{Z} = \text{pr}_{<\mathcal{H}(Y)>}(X) \quad &\Leftrightarrow \quad < X - \hat{Z}, Z >_{\mathcal{H}} = 0 \ \forall \ Z \in \mathcal{H}(Y) \\ &\Leftrightarrow \quad < X - \hat{Z}, Y_i >_{\mathcal{H}} = 0 \ \forall \ i = 1 \ldots n \quad \text{(E.10)} \\ &\doteq \quad \hat{E}[X|Y] \quad \text{(E.11)} \end{aligned}$$

$$\doteq \hat{X}(Y) \tag{E.12}$$

The notation $\hat{E}[X|Y]$ is often used for the projection of X over the span of Y [1].

E.1.2 Solution for the linear (scalar) estimator

Let $Z = AY$ be a linear estimator for the random *variable* $X \in \mathbb{R}$; $A \in T\mathbb{R}^n$ is a row-vector, and $Y \in \mathbb{R}^n$ an n-dimensional column vector. The least-square estimate $\hat{Z}$ is given by the choice of A that solves the following problem:

$$\hat{A} = \arg\min_A \|AY - X\|^2_{\mathcal{H}} \tag{E.13}$$

where $\|\cdot\|_{\mathcal{H}} = E\|\cdot\|$ is the norm induced by the inner product $<\cdot,\cdot>_{\mathcal{H}}$.

Theorem E.1.1 *The solution $\hat{Z} = \hat{A}Y$ to the problem (E.13) exists, is unique and corresponds to the orthogonal projection of X onto the span of Y:*

$$\hat{Z} = \mathrm{pr}_{<\mathcal{H}(Y)>}(X) \tag{E.14}$$

The proof is an easy exercise. In the following we report an explicit construction of the best estimator $\hat{A}$. From substituting the expression of the estimator onto the definition of orthogonal projection (E.12), we get

$$0 =< X - \hat{A}Y, Y_i >_{\mathcal{H}} = E[(X - \hat{A}Y)Y_i] \tag{E.15}$$

which holds iff $EXY_i = \hat{A}EYY_i \; \forall \; i = 1 \ldots n$. In a row-vector notation we write

$$\begin{aligned} EXY^T &= \hat{A}EYY^T \\ \Sigma_{XY} &= \hat{A}\Sigma_Y \end{aligned} \tag{E.16}$$

which, provided that $\mathcal{H}(Y)$ is full rank, gives $\hat{A} = \Sigma_{XY}\Sigma_Y^{-1}$.

[1] The resemblance with a conditional expectation is due to the fact that, in the presence of Gaussian random vectors such a projection is indeed the conditional expectation.

E.1.3 Affine least-variance estimator

Suppose we want to compute the best estimator of a zero-mean *random vector* X as a *linear* map of the zero-mean random vector Y. We just have to repeat the construction reported in the previous section for each component X_i of X, so that the rows $\hat{A}_{i.}$ of the matrix $\hat{A}$ are given by

$$\begin{aligned} \hat{A}_{1.} &= \Sigma_{X_i Y}\Sigma_Y^{-1} \\ \vdots &= \vdots \\ \hat{A}_{n.} &= \Sigma_{X_n Y}\Sigma_Y^{-1} \end{aligned} \tag{E.17}$$

which eventually gives us

$$\hat{A} = \Sigma_{XY}\Sigma_Y^{-1}. \tag{E.18}$$

If now the vectors X and Y are not zero-mean, $\mu_X \neq 0$, $\mu_Y \neq 0$, we first transform it into a zero-mean problem by defining $\bar{Y} \doteq Y - \mu_Y$, $\bar{X} \doteq X - \mu_X$, then solve for the linear least-variance estimator $\hat{A} = \Sigma_{\bar{X}\bar{Y}}\Sigma_{\bar{Y}}^{-1} \doteq \Sigma_{XY}\Sigma_Y^{-1}$, and then substitute to get

$$\hat{Z} = \mu_X + \Sigma_{XY}\Sigma_Y^{-1}(Y - \mu_Y) \tag{E.19}$$

which is the least-variance affine estimator

$$\hat{Z} \doteq \hat{E}[X|Y] = \hat{A}Y + \hat{b} \tag{E.20}$$

where

$$\hat{A} = \Sigma_{XY}\Sigma_Y^{-1} \tag{E.21}$$

$$\hat{b} = \mu_X - \Sigma_{XY}\Sigma_Y^{-1}\mu_Y. \tag{E.22}$$

It is an easy exercise to compute the variance of the estimation error $\tilde{X} \doteq X - \hat{Z}$:

$$\Sigma_{\tilde{X}} = \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}. \tag{E.23}$$

If we interpret the variance of X as the "prior uncertainty", and the variance ot $\tilde{X}$ as the "posterior uncertainty", we may interpret the second term (which is positive semi-definite) of the above equation as a "decrease" of the uncertainty.

E.1.4 Properties and interpretations of the least-variance estimator

The variance of the estimation error in equation (E.23) is by construction the smallest that can be achieved *with an affine estimator.* Of course if we consider a broader class $\mathcal{T}$ of estimators, the estimation error can be further decreased, unless the model that generates the data T is itself affine:

$$Y = T(X) = FX + W. \tag{E.24}$$

In such a case, using the matrix inversion lemma [2], it is easy to compute the expression of the optimal (affine) estimator that depends only upon Σ_X, Σ_W and F:

$$\hat{Z} = \Sigma_X F^T (F\Sigma_X F^T + \Sigma_W)^{-1} Y \tag{E.25}$$

which achieves a variance of the estimation error equal to

$$\Sigma_{\tilde{X}} = \Sigma_X - \Sigma_X F^T (F\Sigma_X F^T + \Sigma_W)^{-1} F\Sigma_X. \tag{E.26}$$

Projection onto an orthogonal sum of subspaces

Let $Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$ be such that

$$\mathcal{H}(Y) = \mathcal{H}(Y_1) \oplus \mathcal{H}(Y_2). \tag{E.27}$$

We may now wonder what are the conditions under which

$$\hat{E}[X|Y] = \hat{E}[X|Y_1] + \hat{E}[X|Y_2]. \tag{E.28}$$

[2] If A, B, C, D are real matrices of the appropriate dimensions with A and C invertible, then $(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA$.

After an easy calculation one can see that the above is true iff $EY_1Y_2^T = 0$, which is to say when

$$\mathcal{H}(Y_1) \perp \mathcal{H}(Y_2) \tag{E.29}$$

Change of basis

Suppose that instead of measuring the instances of a random vector Y we measure another random vector Z which is related to Y via a change of basis: $Z = TY \mid T \in \mathcal{GL}(m)$. If we call $\hat{E}[X|Y] = \hat{A}Y$, then it is immediate to see that

$$\begin{aligned} \hat{E}[X|Z] &= \Sigma_{XZ}\Sigma_Z^{-1}Z \\ &= \Sigma_{XY}T^T(T^{T^{-1}}\Sigma_Y T^{-1})Z \\ &= \Sigma_{XY}\Sigma_Y^{-1}T^{-1}Z. \end{aligned} \tag{E.30}$$

Innovations

The linear least-variance estimator involves the computation of the inverse of the output covariance matrix Σ_Y. It may be interesting to look for changes of bases T that transform the output Y into $Z = TY$ such that $\Sigma_Z = I$. In such a case the optimal estimator is simply

$$\hat{E}[X|Z] = \Sigma_{XZ}Z. \tag{E.31}$$

Let us pretend for a moment that the components of the vector Y are samples of a process taken over time: $Y_i = y(i)$, and call $y^t = [Y_1, \ldots, Y_t]^T$ the history of the process up to time t. Each component (sample) is an element of the Hilbert space $\mathcal{H}$, which has a well-defined notion of orthogonality, and where we can apply Gram-Schmidt procedure in order to make the "vectors" $y(i)$ orthogonal (uncorrelated).

$$\begin{array}{lcccl} v_1 & \doteq & y(1) & \longrightarrow & e_1 \doteq \frac{v_1}{\|v_1\|} \\ v_2 & \doteq & y(2) - < y(2), e_1 > e_1 & \longrightarrow & e_2 \doteq \frac{v_2}{\|v_2\|} \\ \vdots & \doteq & \vdots & \longrightarrow & \vdots \\ v_t & \doteq & y(t) - \sum_{i=1}^{t-1} < y(i), e_i > e_i & \longrightarrow & e_t \doteq \frac{v_t}{\|v_t\|} \end{array}$$

The process $\{e\}$, whose instances up to time t are collected into the vector $e^t = [e_1, \ldots, e_t]^T$ has a number of important properties:

1. The component of e^t are *orthonormal* in $\mathcal{H}$ (or equivalently $\{e\}$ is an uncorrelated process). This holds by construction.

2. The transformation from y to e is *causal*, in the sense that – if we represent it as a matrix L_t such that

$$y^t = L_t e^t \tag{E.32}$$

then $L_t \in \mathcal{L}_+$ is lower-triangular with positive diagonal. This follows from the Gram-Schmidt procedure.

3. The process $\{e\}$ is *equivalent* to $\{y\}$ in the sense that they generate the same span

$$\mathcal{H}(y^t) = \mathcal{H}(e^t). \tag{E.33}$$

This property follows from the fact that L_t is non-singular.

4. If we write $y^t = L_t e^t$ in matrix form as $Y = LE$, then $\Sigma_Y = LL^T$.

The meaning of the components of v, and the name *innovation*, comes from the fact that we can interpret

$$v_t \doteq y(t) - \hat{E}[y(t)|y^{t-1}] \tag{E.34}$$

as a *one-step prediction error*. The process e is a scaled version of v such that its variance is the identity.

We may now wonder whether each process $\{y\}$ has an innovation, and if so, whether it is unique. The following theorem, which is known as *Choleski factorization theorem* or *Spectral Factorization theorem* depending upon the context, states the conditions:

Theorem E.1.2 *There exists a unique vector E which is causally equivalent to Y iff there exists a unique lower-triangular matrix L, called Choleski's factor, such that $\Sigma_Y = LL^T$.*

Remark E.1.1 *The Choleski factor can be interpreted as a "whitening filter", in the sense that it acts on the components of the vector Y in a causal fashion to make them uncorrelated.*

We may consider a two-step solution to the problem of finding the least-square filter: a "whitening step"

$$E = L^{-1}Y \tag{E.35}$$

where $\Sigma_E = I$, and a projection onto $\mathcal{H}(E)$:

$$\hat{X}(Y) = \Sigma_{XE}L^{-1}Y. \tag{E.36}$$

E.2 Linear least-variance estimator for stationary processes

In the previous section we have interpreted a column-vector as a collection of samples from a scalar random process, and computed the least-variance estimator by orthogonal projection. In this section we see how this plot generalizes to proper *stationary* processes. We consider only scalar processes for simplicity of notation, although all considerations can be extended to vector-valued processes.

Let us assume that $\{x(t)\} \in \mathbb{R}^n$ and $\{y(t)\} \in \mathbb{R}^m$ are (wide-sense) jointly stationary, i.e.

$$\Sigma_{xy}(t,s) \doteq Ex(t)y^T(s) = \Sigma_{xy}(t-s). \tag{E.37}$$

Again, we restrict our attention to *linear* estimators of $\{x(t)\}$ given the measurements of $\{y(s);\ s \leq t\}$ *up to time* t. We denote the estimate by $\hat{x}(t|t)$. A linear estimator is described by a convolution kernel h such that

$$\hat{x}(t|t) = \sum_{k=-\infty}^{t} h(t,k)y(k). \tag{E.38}$$

The design of the least-variance estimator involves finding the kernel $\hat{h}$ such that the estimation error $\tilde{x}(t) \doteq x(t) - \hat{x}(t|t)$ has minimum variance. This is found, as in the previous sections for the static case, by imposing that the estimation error be orthogonal to the history of the process $\{y\}$ up to time t:

$$\begin{aligned} < x(t) - \hat{x}(t|t), y(s) >_{\mathcal{H}} &= 0 \ \forall\ s \leq t \\ Ex(t)y^T(s) - \sum_{k=-\infty}^{t} h(t,k)Ey(k)y^T(s) &= 0 \ \forall\ s \leq t \end{aligned} \tag{E.39}$$

which is equivalent to

$$\Sigma_{xy}(t-s) = \sum_{k=-\infty}^{t} h(t,k)\Sigma_y(k-s). \tag{E.40}$$

The above is equivalent to a linear system with an infinite number of equations, and we will assume that it has a unique solution for H. Given that the processes involved are (jointly) stationary, and the convolution starts at $-\infty$, it can be easily seen that the kernel h is *time invariant*: $h(t,k) = h(t-k)$. Therefore the last equation is equivalent to

$$\Sigma_{xy}(t) = \sum_{s=0}^{\infty} h(s)\Sigma_y(t-s) \ \forall t \geq 0 \tag{E.41}$$

which is called *Wiener-Hopf equation* and is exactly equivalent to the orthogonality conditions (E.16). In fact, if we $\mathcal{Z}$-transform the above equation

$$S_{xy}(z) = H(z)S_y(z) \tag{E.42}$$

we have exactly the same expression as equation (E.16), which we could try to solve as

$$\hat{H}(z) = S_{xy}(z)S_y^{-1}(z) \tag{E.43}$$

provided that the spectral density S_y is invertible. This, however, is not quite the solution we are looking for. In fact, in order to be of any use, the estimator must be *causal* (it must not depend upon "future" samples of the process $\{y\}$) and *stable* (it must return a bounded estimate for bounded data). We can express these conditions by requiring

- causality: $h(t) = 0 \ \forall t < 0$ (or $H(z)$ analytic at ∞)
- stability: $H(z)$ analytic in $|z| \geq 1$ (or $h(t)$ square-summable).

One particular case is when the spectral density of $\{y\}$ is the identity (or equivalently $\{y\}$ is a white noise). Then $S_y = I$ and we could choose

$$h(t) = \begin{cases} \Sigma_{xy}(t) & t \geq 0 \\ 0 & t < 0. \end{cases} \tag{E.44}$$

This suggests us to try to *whiten* (or orthonormalize) the measurement process $\{y\}$ in a similar fashion to

what we did in section E.1.4. Indeed we can state a theorem similar to E.1.2, which is known as the *spectral factorization theorem*:

Theorem E.2.1 *There exists a process* $\{\tilde{e}\}$ *such that* $\mathcal{H}(\tilde{e}^t) = \mathcal{H}(y^t)$ *and* $\Sigma_{\tilde{e}}(t) = \Lambda\delta(t)$ *iff there exists* $W(z)$ *stable and causal, with* $W^{-1}(z)$ *causal such that* $S_y(z) = W(z)W(z^{-1})$.

Remark E.2.1 *In words there exists a white process* $\{\tilde{e}\}$ *(called the* innovation*) which is causally equivalent to* $\{y\}$ *iff the spectral density of* y *has a causal, stable and minimum-phase spectral factor. If we re-scale* $W(z)$ *to* $L(z) = W(z)W(\infty)^{-1}$, *the innovation* $\{e\}$ *is re-normalized so that* $\Sigma_e(t) = I\delta(t)$, *and is called* normalized innovation.

We may at this point repeat the two-step construction of the least-variance estimator. First the "whitening step":

$$E(z) = L^{-1}(z)Y(z) \tag{E.45}$$

and then the *causal* part of the projection:

$$\hat{X}(Y) = \begin{cases} \Sigma_{xe}(t) * e(t) & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{E.46}$$

where $*$ indicates the standard convolution. Equivalently, if we denote by $[S_{xe}(z)]_+$ the causal part of the $\mathcal{Z}$-transform of $\Sigma_{xe}(t)$, we can write

$$\hat{X}(Y)_{(z)} = [S_{xe}(z)]_+ E(z). \tag{E.47}$$

Since $S_{xe}(z) = S_{xy}(z)L(z^{-1})^{-1}$, the final expression of our linear, least-variance estimator is (in the $\mathcal{Z}$-domain) $\hat{x} = \hat{H}(z)y(z)$, where the kernel H is given by

$$\hat{H}(z) = \left[S_{xy}(z)L^{-1}(z^{-1})\right]_+ L^{-1}(z). \tag{E.48}$$

The corresponding filter is known as the *Wiener filter*. Again we can recover the meaning of the innovation as the one-step prediction error for the measurements: in fact, the best prediction of the process $\{y\}$, indicated with $\hat{y}(t|t-1)$, is defined as the projection of $y(t)$ onto the span of $\{y\}$ up to $t-1$, indicated with $\mathcal{H}_{t-1}(y)$.

Such projection is therefore defined such that

$$y(t) = \hat{y}(t|t-1) \overset{\perp}{+} e(t) \tag{E.49}$$

where $e(t) \perp \mathcal{H}_{t-1}(y) = \mathcal{H}_{t-1}(e)$.

E.3 Linear, finite-dimensional stochastic processes

A linear, finite-dimensional stochastic process (LFDSP) is defined as the output of a linear, finite-dimensional dynamical system driven by white Gaussian noise. Let $A(t)$, $B(t)$, $C(t)$, $D(t)$ be time-varying matrices of suitable dimensions, $\{n(t)\} \in \mathcal{N}(0, I) \mid En(t)n^T(s) = I\delta(t-s)$ a white, zero-mean Gaussian noise and x_0 a random vector which is uncorrelated with $\{n\}$: $Ex_0n^T(t) = 0 \ \forall \ t$. Then $\{y(t)\}$ is a LFDSP if there exists $\{x(t)\}$ such that

$$\begin{cases} x(t+1) = A(t)x(t) + B(t)n(t) & x(t_0) = x_0 \\ y(t) = C(t)x(t) + D(t)n(t) \end{cases} \tag{E.50}$$

We call $\{x\}$ the *state process*, $\{y\}$ the *output (or measurement) process*, and $\{n\}$ the *input (or driving) noise.* The time-evolution of the state process can be written as the orthogonal sum of the past history (prior to the initial condition), and the present history (from the initial condition until the present time)

$$x(t) = \Phi_{t_0}^t x_0 + \sum_{k=t_0}^{t-1} \Phi_{k+1}^t B(t)n(t) = \hat{E}[x(t)|\mathcal{H}(x^{t_0})] \overset{\perp}{+} \hat{E}[x(t)|x(t_0), \ldots, x(t-1)] \tag{E.51}$$

where Φ denotes a fundamental set of solutions, which is the flow of the differential equation

$$\begin{cases} \Phi(t+1, s) = A(t)\Phi(t, s) \\ \Phi(t, t) = I. \end{cases} \tag{E.52}$$

In the case of a time-invariant system $A(t) = A \ \forall t$, then $\Phi(t, s) = A^{t-s}$.

Remark E.3.1 *As a consequence of the definitions, the orthogonality between the state and the input noise propagates up to the current time:*

$$n(t) \perp_{\mathcal{H}} x(s) \ \forall s \leq t. \tag{E.53}$$

Moreover, the past history up to time s is alway summarized by the value of the state at that time (Markov property):

$$\hat{E}[x(t)|\mathcal{H}_s(x)] = \hat{E}[x(t)|x(s)] = \Phi(t,s)x(s) \ \forall t \geq s. \tag{E.54}$$

E.4 Stationariety of LFDSP

In order to design the least-squares estimator as in the previous sections, we ask what are the conditions under which a LFDSP is stationary. The first restriction we require is that the system be time-invariant. The mean of the state process at time t is given by

$$\mu_x(t) = A^{t-t_0}\mu_{x_0} \tag{E.55}$$

while the covariance of the state-process

$$\Sigma_x(t,s) = A^{t-s}\Sigma_x(s) \tag{E.56}$$

evolves according to the following *Ljapunov equation*

$$\Sigma_x(s+1) = A\Sigma_x(s)A^T + BB^T. \tag{E.57}$$

The conditions for stationariety impose that $\sigma_x(t) = \text{const}$ and $\mu_x(t) = \text{const}$. It is easy to prove the following

Theorem E.4.1 *Let A be stable (have all eigenvalues in the unit complex circle), then* $\Sigma_x(t-t_0) \rightarrow \bar{\Sigma}$, *where* $\bar{\Sigma} = \sum_{k=0}^{\infty} A^k BB^T A^{T^k}$ *is the unique equilibrium solution of the above Ljapunov equation, and* $\{x\}$ *describes asymptotically a stationary process. If* x_0 *is such that* $\Sigma_x(t_0) = \bar{\Sigma}$, *then the process is stationary* $\forall\ t \geq t_0$.

Remark E.4.1 *The condition of stability for A is sufficient, but not necessary for generating a stationary process. If, however, the pair* (A, B) *is completely controllable, so that the noise input affects all of the components of the state, then such a stability condition becomes also necessary.*

E.5 The linear Kalman filter

Suppose we are given a linear finite-dimensional process, which has a realization (A, B, C, D) as in equation (E.50). While we measure the (noisy) output $y(t)$ of such a realization, we do not have access to its state $x(t)$. The Kalman filter is a dynamical model that accepts as input the output of the process realization, and returns an estimate of its state that has the property of having the least error variance. In order to derive the expression for the filter, we write the LFDSP as follows:

$$\begin{cases} x(t+1) = Ax(t) + v(t) & x(t_0) = x_0 \\ y(t) = Cx(t) + w(t) \end{cases} \tag{E.58}$$

where we have neglected the time argument in the matrices $A(t)$ and $C(t)$ (all considerations can be carried through for time-varying systems as well). $v(t) = Bn(t)$ is a white, zero-mean Gaussian noise with variance Q, $w(t) = Dn(t)$, also a white, zero-mean noise, has variance R, so that we could write

$$\begin{aligned} v(t) &= \sqrt{Q}n(t) \\ w(t) &= \sqrt{R}n(t) \end{aligned}$$

where n is a unit-variance noise. In general v and w will be correlated, and in particular we will call

$$S(t) = E[v(t)w^T(t)]. \tag{E.59}$$

We require that the initial condition x_0 be uncorrelated from the noise processes:

$$x_0 \perp \{v\}, \{w\} \; \forall \; t \tag{E.60}$$

The first step is to modify the above model so that the model error v is uncorrelated from the measurement error w.

Uncorrelating the model from the measurements

In order to uncorrelate the model error from the measurement error we can just substitute v with the complement of its projection onto the span of w. Let us call

$$\tilde{v}(t) = v(t) - \hat{E}[v(t)|H(w)] = v(t) - \hat{E}[v(t)|w(t)] \tag{E.61}$$

the last equivalence is due to the fact that w is a white noise. We can now use the results from section E.1 to conclude that

$$\tilde{v}(t) = v(t) - SR^{-1}w(t) \tag{E.62}$$

and similarly for the variance matrix

$$\tilde{Q} = Q - SR^{-1}S^T. \tag{E.63}$$

Substituting the expression of $v(t)$ into the model (E.58) we get

$$\begin{cases} x(t+1) = Fx(t) + SR^{-1}y(t) + \tilde{v} \\ y(t) = Cx(t) + w(t) \end{cases} \tag{E.64}$$

where $F = A - SR^{-1}C$. The model error $\tilde{v}$ in the above model is uncorrelated from the measurement noise w, and the cost is that we had to add an output-injection term $SR^{-1}y(t)$.

Prediction step

Suppose at some point in time we are given a current estimate for the state $\hat{x}(t|t)$ and a corresponding estimate of the variance of the model error $P(t|t) = E[\tilde{x}(t)\tilde{x}(t)^T]$ where $\tilde{x} = x - \hat{x}$. At the initial time t_0 we can take $\hat{x}(t_0|t_0) = x_0$ with some bona-fide variance matrix. Then it is immediate to compute

$$\hat{x}(t+1|t) = F\hat{x}(t|t) + SR^{-1}y(t) + \hat{E}[\tilde{v}(t)|H_t(y)] \tag{E.65}$$

where the last term is zero since $\tilde{v}(t) \perp x(s) \forall x \leq t$ and $\tilde{v}(t) \perp w(s) \forall s$ and therefore $\tilde{v}(t) \perp y(s) \forall s \leq t$. The estimation error is therefore

$$\tilde{x}(t+1|t) = F\tilde{x}(t|t) \overset{\perp}{+} \tilde{v}(t) \tag{E.66}$$

where the sum is an orthogonal sum, and therefore it is trivial to compute the variance as

$$P(t+1|t) = FP(t|t)F^T + \tilde{Q}. \tag{E.67}$$

Update step

Once a new measurement is acquired, we can update our prediction so as to take into account the new measurement. The update is defined as $\hat{x}(t+1|t+1) \doteq \hat{E}[x(t+1)|H_{t+1}(y)]$. Now, as we have seen in section E.1.4, we can decompose the span of the measurements into the orthogonal sum

$$H_{t+1}(y) = H_t(y) \overset{\perp}{+} \{e(t+1)\} \tag{E.68}$$

where $e(t+1) \doteq y(t+1) - \hat{E}[y(t+1)|H_t(y)]$ is the innovation process. Therefore, we have

$$\hat{x}(t+1|t+1) = \hat{E}[x(t+1)|H_t(y)] + \hat{E}[x(t+1)|e(t+1)] \tag{E.69}$$

where the last term can be computed using the results from section E.1:

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + L(t+1)e(t+1) \tag{E.70}$$

where $L(t+1) \doteq \Sigma_{xe}(t+1)\Sigma_e^{-1}(t+1)$ is called the *Kalman gain.* Substituting the expression for the innovation we have

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + L(t+1)\left(y(t+1) - C\hat{x}(t+1|t)\right) \tag{E.71}$$

from which we see that the update consists in a linear correction weighted by the Kalman gain.

Computation of the gain

In order to compute the gain $L(t+1) \doteq \Sigma_{xe}(t+1)\Sigma_e^{-1}(t+1)$ we derive an alternative expression for the innovation:

$$e(t+1) = y(t+1) - Cx(t+1) + Cx(t+1) - C\hat{x}(t+1|t) = w(t+1) + C\tilde{x}(t+1|t) \tag{E.72}$$

from which it is immediate to compute

$$\Sigma_{xe}(t+1) = P(t+1|t)C^T. \tag{E.73}$$

Similarly we can derive the variance of the innovation $\Lambda(t+1)$:

$$\Lambda(t+1) \doteq \Sigma_e(t+1) = CP(t+1|t)C^T + R \tag{E.74}$$

and therefore the Kalman gain is

$$L(t+1) = P(t+1|t)C^T\Lambda^{-1}(t+1). \tag{E.75}$$

Variance update

From the update of the estimation error

$$\tilde{x}(t+1|t+1) = \tilde{x}(t+1|t) - L(t+1)e(t+1) \tag{E.76}$$

we can easily compute the update for the variance. We first observe that $\tilde{x}(t+1|t+1)$ is by definition orthogonal to $H_{t+1}(y)$, while the correction term $L(t+1)e(t+1)$ is contained in the history of the innovation, which is by construction equal to the history of the process y: $H_{t+1}(y)$. Then it is immediate to see that

$$P(t+1|t+1) = P(t+1|t) - L(t+1)\Lambda(t+1)L^T(t+1). \tag{E.77}$$

The above equation is not convenient for computational purposes, since it does not guarantee that the updated variance is a symmetric matrix. An alternative form of the above that does guarantee symmetry of the result is

$$P(t+1|t) = \Gamma(t+1)P(t+1|t)\Gamma(t+1)^T + L(t+1)RL(t+1)^T \tag{E.78}$$

where $\Gamma(t+1) = I - L(t+1)C$. The last equation is in the form of a discrete Riccati equation (DRE).

Predictor equations

It is possible to combine the two steps above and derive a single model for the one-step predictor. We summarize the result as follows:

$$\hat{x}(t+1|t) = A\hat{x}(t|t-1) + K_s(t)\left(y(t) = C\hat{x}(t|t-1)\right) \tag{E.79}$$

$$P(t+1|t) = F\Gamma(t)P(t|t-1)\Gamma(t)^T F^T + FL(t)RL^T(t)F^T + \tilde{Q} \tag{E.80}$$

where we have defined

$$K_s(t) \doteq FL(t) + SR^{-1} \tag{E.81}$$

$$= \left(AP(t|t-1)C^T + S\right)\Lambda^{-1}(t) \tag{E.82}$$

E.6 Asymptotic properties

If we consider time-invariant models (all matrices A, C, Q, S, R are constant in time), we can study the asymptotic behavior of the estimator.

Remark E.6.1 *In particular, the dynamics of the estimator depends upon $P(t)$, the solution of the DRE of equation (E.78). We want such a solution to converge asymptotically to a small* but non-zero *value. In fact, $P = 0$ corresponds to a zero gain $K = 0$, which indicates that the filter does take into account the measurements. In such a case we say that the filter is* saturated.

We will not get into the details of the results of the asymptotic theory of Kalman filtering. We will only report the main results, which essentially says that if the realization of the LFDSP is minimal, then

there exists a unique positive-definite fixed-point of the DRE, and the solution converges to the fixed-point asymptotically. Furthermore the dynamics of the estimation error is stable (even though the process may be unstable). The Kalman filter converges asymptotically to the Wiener filter described in section E.2.

Claim E.6.1 *If the pair* (F, C) *is detectable and* $(F, \sqrt{Q})$ *is stabilizable, then there exists a unique* $P \mid P = P^T \geq 0$ *fixed point of the DRE (E.78). Furthermore* $P(t) \rightarrow P$ *for all positive semi-definite* $P(t_0)$ *and* $\Gamma = lim_{t\rightarrow\infty}\Gamma(t)$ *is stable.*

We recall that a (F, C) being detectable means that the unobservable subspace is stable, as well as $(F, \sqrt{Q})$ being stabilizable means that the uncontrollable subspace is stable. The proof of the above claim, as well as other results on the asymptotic properties of the Kalman filter, can be found for instance in [55].

Appendix F Observability, observers and identification

F.1 Linear observability

Let us consider a linear system in the form

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) & x(0) = x_0 \\ y(t) = C(t)x(t). \end{cases} \tag{F.1}$$

We say that the above model is *completely observable* if, given pairs $\{y(t), u(t)\}_{t\in[t_0,t_1]}$ on an interval $[t_0, t_1]$, it is possible to reconstruct uniquely the initial condition x_0. Once the initial condition is known we can reconstruct the whole trajectory $x(t)$ by simply integrating the state model. If we call $\Phi(t, t_0)$ the fundamental set of solutions of the differential equation above, i.e. a matrix of the same size of $A(t)$ such that

$$\Phi(t+1, t_0) = A(t)\Phi(t, t_0) \qquad \Phi(t_0, t_0) = I \tag{F.2}$$

then the output $y(t)$ can be written as

$$y(t) = C(t)\Phi(t, t_0)x_0 + C(t)\int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)d\tau. \tag{F.3}$$

From the above expression we can see that the input u is irrelevant, since we may substitute the measurements $y(t)$ with $\tilde{y}(t) = y(t) - C(t)\int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)d\tau$. Therefore, observability is only concerned with $(A(t), C(t))$. For this reason one often speaks about the observability of the pair $(A(t), C(t))$. We call the operator $L_o(t) \doteq C(t)\Phi(t, t_0)$ the *observability operator*. It is easy to prove the following

Claim F.1.1 *The following statements are equivalent:*

- *The pair $(A(t), C(t))$ is completely observable*

- $Nu(L_o) = \{0\}$
- $N_o \doteq L_o L_o^*$ *is non-singular;* L_o^* *denotes the adjoint operator of* L_o.

Note that the last test consists in verifying the rank of the linear finite-dimensional operator N_o. An explicit expression for such an operator, which is called the *observability grammian*, is given by

$$N_o(t, t_0) = \int_{t_0}^{t} \Phi^*(\tau, t) C^*(\tau) C(\tau) \Phi(\tau, t) d\tau. \tag{F.4}$$

If the system is time-invariant, so that the matrices A, B, C are constant, then the fundamental set of solutions of the state integral is simply

$$\Phi(t, t_0) = A^{(t-t_0)} \tag{F.5}$$

and testing for observability reduces to verifying that the matrix

$$O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{F.6}$$

has full-rank. Note that it is sufficient to consider n steps, where n is the dimension of the state-space, as it is easy to see using Cayley-Hamilton theorem. Equivalent characterizations can be easily derived (see for instance [57]):

Claim F.1.2 *The following statements are equivalent:*

- *The linear-time invariant system (F.1) is observable*
- *O has full rank*
- *there exists a matrix L such that the spectrum of $A - LC$ can be assigned arbitrarily*
- *the rank of* $\begin{bmatrix} \lambda I - A \\ C \end{bmatrix}$ *is full for all choices of* $\lambda \in \mathbb{R}$.

F.2 Linear observers

Suppose we are given a linear system in the form (F.1). An observer for it is just another dynamical model that accepts as inputs the input/output pairs of the original system $u(t), y(t)$, and produces an estimate of the state of the original system. The most trivial form of observer consists in a copy of the original model. If equation (F.1) represents some physical process, we may produce a *model* of it (for instance a computer program) driven by the same dynamics, where we have access to the states:

$$\hat{x}(t+1) = A(t)\hat{x}(t) + B(t)u(t) \qquad \hat{x}(t_0) = \hat{x}_0. \tag{F.7}$$

Of course, if the above equation is just an approximation of the actual physical process and there is even a slight error in the initial conditions, the state of the observer could drift arbitrarily away from the state of the original model.

Since all we can measure from the original system is the *output*, we require it to be close to the output that would be produced by the estimated state: $\hat{y}(t) = C(t)\hat{x}(t)$. If this is different from the output $y(t)$ we could use the difference as an error to be fed back to the state, for instance with a linear correction

$$\hat{x}(t+1) = A(t)\hat{x}(t) + B(t)u(t) + L(t)\left(y(t) - C(t)\hat{x}(t)\right) \tag{F.8}$$

where the gain $L(t)$ can be chosen so as to satisfy conditions of stability of the observer, or other optimality conditions. The error $\tilde{x} = x - \hat{x}$ can be easily seen to obey

$$\tilde{x}(t+1) = (A(t) - L(t)C(t))\tilde{x}(t) \tag{F.9}$$

from which we see that if the model is time-invariant and observable we can choose L so that the error decays to zero exponentially.

The Kalman filter is a special type of observer for systems where the input u is a white, zero-mean Gaussian process. In such a case, L is chosen so as to guarantee the minimum variance of the estimation error. While in this section we have postulated a linear update, using heuristic arguments, in [58] it is proven that the optimal filter does have the structure of a linear update as in the above equation, where $L(t)$ is

choosen by solving a discrete Riccati equation.

F.3 Nonlinear observability

In this section we report some notation, referring to [48, 63, 64, 66, 82, 104], for the system:

$$\begin{cases} \dot{x} = f(x,u) \ ; \ x(t_0) = x_0 \\ y = h(x) \end{cases} \qquad (*)$$

where $x \in N \subset \mathbb{R}^n$, some n-dimensional manifold, $u \in M \subset \mathbb{R}^m$ and $y \in P \subset \mathbb{R}^p$; it is assumed that f and h are smooth functions. The set of admissible inputs is described as $\mathcal{U} \doteq \{u : \mathbb{R}^+ \to P \subset \mathbb{R}^p\}$ such that

1. $\mathcal{U}$ is closed under concatenation
2. f describes a family of vector fields parametrized by $\bar{u} \in P$.
3. u are piecewise constant functions which are piecewise continuous from the right:

$$u(t) \doteq \{\bar{u}_i \ for \ t \in I_i = [t_1 + \ldots + t_{i-1}, t_1 + \ldots + t_i) \mid \bar{u}_i \in P \subset \mathbb{R}^p \ , \ \forall i\}.$$

We call $f_i \doteq f(x, \bar{u}_i)$; in the time interval I_i the system evolves along the integral curve of f_i. The above assumptions may be partially released; however, they are general enough for our purposes. In studying the visual motion problem, we will be mostly concerned with the autonomous case: $u(t) = 0 \ \forall t$.

Definition F.3.1 *x_1 and x_2 are said to be* indistinguishable *(and denoted with $x_1 I x_2$)* $\stackrel{*}{\Leftrightarrow} \forall u \in \mathcal{U}$, $h(\phi_t(x_1, u)) = h(\phi_t(x_2, u)) \ \ \forall t \geq 0$.

$I(x) \doteq \{x_i \mid x_i I x \ ; \ x \in N\}$ is the set of states which are indistinguishable from x.

Definition F.3.2 *$(*)$ is completely observable (C-O) at x* $\stackrel{*}{\Leftrightarrow} I(x) = \{x\}$.

$(*)$ is completely observable $\stackrel{*}{\Leftrightarrow}$ *it is C-O at $x \ \forall \ x \in N$.*

Definition F.3.3 *Given an open set $U \subset N$, x_1 and x_2 are said to be* U-indistinguishable *(and denoted with $x_1 I^U x_2$)* $\stackrel{*}{\Leftrightarrow} \{\phi_t(x_1, u) \in U, \phi_t(x_2, u) \in U \ \forall t \in [t_0, t_1]\} \Rightarrow h(\phi_t(x_1, u)) = h(\phi_t(x_2, u)) \ \ \forall t \in [t_0, t_1]$.

$I^U(x) \doteq \{x_i \mid x_i I^U x\}$ is the set of states which are U-indistinguishable from x.

Definition F.3.4 $(*)$ *is said to be locally weakly observable (L-W-O)* at $x \overset{*}{\Leftrightarrow} \exists U, x \in U \mid \forall V \subset U, x \in V,\ I^V(x) = \{x\}$.

$(*)$ *is said to be* locally weakly observable $\overset{*}{\Leftrightarrow}$ *it is L-W-O at* $x\ \forall x \in N$.

Definition F.3.5 *The* observability space $\mathcal{O}$ *for* $(*)$ *is defined to be the smallest subspace of* $C^\infty(N)$ *which contains the functions* $\{h_1 \dots h_p\}$ *and is invariant under Lie differentiation along vector fields in* $\tau \doteq \{f_i = f(x, \bar{u}_i)\}$.

Definition F.3.6 *The* observability codistribution *is defined as*

$$d\mathcal{O} \doteq \{d\lambda \mid \lambda \in \mathcal{O}\}$$

The observability codistribution is the smallest codistribution which is invariant for $(*)$ and contains the forms dh. It can be shown that the definition does not change if we allow the vector fields in τ to belong to the accessibility algebra, which consists of repeated Lie brackets of vector fields in τ.

Definition F.3.7 *A system is said to satisfy the observability rank condition (ORC) at* $p \overset{*}{\Leftrightarrow} dim(d\mathcal{O})_p = n$.

Remark F.3.1 *The ORC can be stated in terms of exterior differential systems. In fact we may interpret the observability codistribution as a Pfaffian system [16]*

$$d\mathcal{O} \doteq dh + dL_{\bar{f}}h + \ldots + dL_{\bar{f}}^{(n-1)}h$$

where $\bar{f} \doteq f(\cdot, \bar{u})$, n *is the dimension of the state-space manifold* N. *The observability rank condition may be state as:*

Definition F.3.8 .

The system $(*)$ *satisfies the observability rank condition at* $p \overset{*}{\Leftrightarrow}\ d\mathcal{O}_p = T_p^*N$

Theorem F.3.1 *If dim* $(\mathcal{O}) = n$ *at* p, *then* $(*)$ *is locally weakly observable in a neighborhood of* p.

Proof:

see [83, 53, 20] This condition is not necessary [20]; however, the following result holds:

Theorem F.3.2 *If $\mathcal{O}$ has constant dimension and the system $(*)$ is locally weakly observable, then rank $(\mathcal{O}) = n$.*

F.4 Identification as a filtering problem

Suppose $\{x(t)\} \in \mathbb{R}^N$ is a trajectory on a linear state-space, which is subject to an implicit dynamic constraint of the form

$$h\left[x(t), dx(t), a\right] = 0 \qquad x(0) = x_0 \qquad a \in M \tag{F.10}$$

where a are some unknown parameters which may move (slowly) on some topological manifold M. Call $\alpha \doteq \psi(a) \in \mathbb{R}^m$ the local coordinates correspondent of a. Suppose we are able to measure x up to some white, zero-mean Gaussian noise:

$$y(t) = x(t) + n(t) \qquad\qquad n \in \mathcal{N}(0, R_n).$$

We are interested in identifying the parameters a recursively from the measurements $\{y(t)\}$ based on the minimization of some cost function of the prediction error (for a classical treatment of prediction error methods (PEM) for linear explicit models see for example [94, 72, 71]).

A common paradigm for PEM identification consists in forcing a Kalman Filter to work as a parameter estimator. The state of the filter is augmented with the unknown parameters, which are described using a random walk model. In this section we will extend this paradigm to nonlinear implicit dynamics and parameters living on a topological manifold. We will restrict our attention to discrete time dynamics, although the same analysis may be carried out for continuous time models.

First we proceed in analogy with the linear-explicit case: we describe the local coordinates of the parameters as first-order random walk, and use the dynamic constraint as an implicit measurement constraint:

$$\begin{cases} \alpha(t+1) = \alpha(t) + n_\alpha(t) & \alpha(0) = \alpha_0 \\ h\left[y(t) - n(t), y(t-1) - n(t-1), \psi^{-1}(\alpha(t))\right] = 0 \end{cases} \tag{F.11}$$

where we have substituted the index t with $t-1$ in the measurements $\{y\}$ (or equivalently the estimator runs with one step delay). We assume n_α, the noise driving the random walk, to be white zero-mean and Gaussian;

its variance R_α may be regarded as a tuning parameter. The noise process $\{n(t)\}$ induces a residual in the measurement equation: if we approximate $x(t)$ with $y(t)$, in general we will observe $h\left[y(t), y(t-1), a\right] = \tilde{n} \neq 0$, where $\tilde{n}$ depends on $\{n\}\{y\}$ and a. This residual – as we will see – is the prediction error (or pseudo-innovation) when choosing a least-squares criterion in the PEM.

Let us collect the measurements into a vector $\bar{y}(t) \doteq \begin{bmatrix} y^T(t) & y^T(t-1) \end{bmatrix}^T$, and similarly with $\bar{n}(t) \doteq [n^T(t)\ n^T(t-1)]^T$. Our task is to estimate α from the model

$$\begin{cases} \alpha(t+1) = \alpha(t) + n_\alpha(t) & \alpha(0) = \alpha_0 \\ h\left[\bar{y}(t) - \bar{n}(t), \psi^{-1}(\alpha(t))\right] = 0. \end{cases} \tag{F.12}$$

In order to follow the course of the linear-explicit case, we have to solve a number of problems:

1. the noise $\bar{n}$ is not white: $E[\bar{n}(t)\bar{n}^T(s)] = \begin{bmatrix} R_n\delta(t-s) & R_n\delta(t-s+1) \\ R_n\delta(t-s-1) & R_n\delta(t-s) \end{bmatrix}$
2. the error $\bar{n}$ does not appear additively in the measurement equation
3. the measurement equation is nonlinear and implicit.

The Extended Kalman Filter (EKF) [58, 17, 55] is a general-purpose local extension to nonlinear systems of the traditional Kalman Filter. It is based on a variational model about the best current trajectory. The system is linearized at each step around the current estimate in order to calculate a correcting gain; the update of the previous estimate is then performed on the original (nonlinear) equations. In order to solve step 3 we need to further extend the EKF to cope with the implicit measurement constraint. This is done in section F.5. We call the result Implicit Extended Kalman Filter (IEKF); some variations of the scheme have been used in different applications in the past years, see for example [23, 26, 46, 33]. The derivation is based on the simple fact that the variational model about the current trajectory is *linear and explicit*, so that the a pseudo-innovation process may be defined analogously to the explicit case.

The derivation of the IEKF in section F.5 does not address the fact that the noise $\bar{n}$ is correlated (see point 2 above). The residual of the measurement equation $\tilde{n}$, which is in fact the pseudo-innovation of the filter, is characterized in terms of $\bar{n}$, provided that the last is white, zero-mean and uncorrelated with n_α. In the following section we will show how to whiten $\bar{n}$ and therefore reduce the problem to a form suitable for using the IEKF as derived in section F.5. Later on we will see how the problem simplifies by assuming

that $\bar{n}$ is white.

F.4.1 Uncorrelating the model from the measurements

Consider a first-order expansion of the measurement equation about the point $\bar{y}(t), \alpha(t)$:

$$h\left[\bar{y}(t), \psi^{-1}(\alpha(t))\right] - D_+(t)n(t) - D_-(t)n(t-1) = \mathcal{O}(\|\bar{n}\|^2) \cong 0$$

where the limit implicit in $\mathcal{O}$ is intended in the mean-square sense, and where we have defined

$$D_+(t) \doteq \left(\frac{\partial h[x(t), x(t-1), a]}{\partial x(t)}\right)_{|\bar{y}(t), \psi^{-1}(\alpha(t))} \tag{F.13}$$

$$D_-(t) \doteq \left(\frac{\partial h[x(t), x(t-1), a]}{\partial x(t-1)}\right)_{|\bar{y}(t), \psi^{-1}(\alpha(t))}. \tag{F.14}$$

Here the residual $\tilde{n}(t) = -D_+(t)n(t) - D_-(t)n(t-1)$ is clearly correlated. In order to estimate the dynamics of $n(t)$, we may insert it into the state: call $z(t) \doteq n(t-1)$.

$$\begin{cases} \alpha(t+1) = \alpha(t) + n_\alpha(t) & \alpha(0) = \alpha_0 \\ z(t+1) = n(t) & z(0) = 0 \\ 0 = h\left[\bar{y}(t), \psi^{-1}(\alpha(t))\right] - D_-(t)z(t) + w(t) \end{cases} \tag{F.15}$$

where we have defined $w(t) \doteq -D_+(t)n(t)$. Now the measurement error w is white; however, it is correlated with the model error $v \doteq [n_\alpha^T, n^T]^T$. We may therefore project the model error onto the span of the measurement error, $H(w)$, in order to make the two orthogonal. We define $\tilde{v}(t) \doteq v(t) - \hat{E}[v(t)|H(w)]$. Since $w(t), n(t)$ and $n_\alpha(t)$ are white, it is easily seen that $\hat{E}[v(t)|H(w)] = \hat{E}[v(t)|w(t)] = E[v(t)w^T(t)]\left(E[w(t)w^T(t)]\right)^{-1} w(t) \doteq \Sigma_{vw}\Sigma_w^{-1}w(t)$. If we define

$$Q(t) \doteq \begin{bmatrix} R_\alpha & 0 \\ 0 & R_n \end{bmatrix} \tag{F.16}$$

$$R(t) \doteq D_+(t)R_n(t)D_+^T(t) \tag{F.17}$$

$$S(t) \doteq \begin{bmatrix} 0 \\ -R_n(t)D_+^T(t) \end{bmatrix} \tag{F.18}$$

it is easy to see that $\Sigma_{vw}\Sigma_w^{-1} = S(t)R^{-1}(t)$; furthermore $\Sigma_{\tilde{v}} \doteq \tilde{Q}(t) = Q(t) + S(t)R^{-1}(t)S^T(t)$. Now $\tilde{v} \doteq v - SR^{-1}w$ is by construction orthogonal (uncorrelated) to w.

F.4.2 A model for PEM identification of nonlinear implicit models

In the previous paragraph we have derived an extended model (up to first-order) with the model error uncorrelated from the measurement error:

$$\begin{cases} \alpha(t+1) = \alpha(t) + n_\alpha(t) & \alpha(0) = \alpha_0 \\ z(t+1) = K(t)\left(h\left[\bar{y}(t), \psi^{-1}(\alpha(t))\right] - D_-(t)z(t)\right) + n(t) & z(0) = 0 \\ 0 = h\left[\bar{y}(t), \psi^{-1}(\alpha(t))\right] - D_-(t)z(t) + w(t) \end{cases} \tag{F.19}$$

where we have defined

$$K(t) \doteq R_n(t)D_+^T(t)\left(D_+(t)R_n(t)D_+^T(t)\right)^{-1} \tag{F.20}$$

$$w(t) \doteq -D_+(t)n(t). \tag{F.21}$$

By applying the results of section F.5, we can derive a pseudo-optimal PEM identification scheme described by the following iteration:

Prediction step

$$\begin{cases} \hat{\alpha}(t+1|t) = \hat{\alpha}(t|t) & \hat{\alpha}(0|0) = \alpha_0 \\ \hat{z}(t+1|t) = K(t)\left(h\left[\bar{y}(t), \hat{\alpha}(t|t)\right] - D_-(t)\hat{z}(t|t)\right) & \hat{z}(0|0) = 0 \\ P(t+1|t) = F(t)P(t|t)F^T(t|t) + \tilde{Q}(t) & P(0|0) = P_0 \end{cases} \tag{F.22}$$

where $F \doteq \begin{bmatrix} I & 0 \\ K(t)\left([C(t) & -D_-(t)]\right) \end{bmatrix}$ and $C(t) \doteq \left(\frac{\partial h[\bar{y}, \psi^{-1}(\alpha)]}{\partial \alpha}\right)_{|\hat{\alpha}(t|t), \bar{y}(t)}$.

Update step

$$\begin{cases} \begin{bmatrix} \hat{\alpha}(t+1|t+1) \\ \hat{z}(t+1|t) \end{bmatrix} = \begin{bmatrix} \hat{\alpha}(t+1|t) \\ z(t+1|t+1) \end{bmatrix} + L(t+1)\left(h\left[\bar{y}(t), \hat{\alpha}(t+1|t)\right] - D_-(t+1)\hat{z}(t+1|t)\right) \\ P(t+1|t+1) = \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + L(t+1)D_+(t+1)R_n(t+1)D_+^T(t+1)L^T(t+1) \end{cases} \tag{F.23}$$

where

$$L(t+1) \doteq P(t+1|t)C^T(t+1)\Lambda^{-1}(t+1) \tag{F.24}$$

$$\Lambda(t+1) \doteq C(t+1)P(t+1|t)C^T(t+1) + D_+(t+1)R_n(t+1)D_+^T(t+1) \tag{F.25}$$

$$\Gamma(t+1) \doteq I - L(t+1)C(t+1) \tag{F.26}$$

Note that we are trying to estimate a process $\{z(t)\}$ which is nearly white noise ($n(t)$ is correlated only within one step). Furthermore, if we expect a large number of measurements, the cost in updating a large state and tuning a large number of model-variance parameters may be relevant. In practical applications the approximation $\bar{n}$ as white noise are often better behaved. In the following section we show how the structure of the filter simplifies under such an approximation.

F.4.3 A simplified version: approximate least-squares PEM identification

In this section we report the equation of the parameter estimator which are obtained supposing that the residual $\tilde{n}$ is white. This corresponds to applying the results of section F.5 directly to the model of eq. (F.12), assuming that $\{\bar{n}\}$ is a white process:

Prediction step

$$\begin{cases} \hat{\alpha}(t+1|t) = \hat{\alpha}(t|t) & \hat{\alpha}(0|0) = \alpha_0 \\ P(t+1|t) = P(t|t) + R_\alpha(t) & P(0|0) = P_0 \end{cases} \tag{F.27}$$

Update step

$$\begin{cases} \hat{\alpha}(t+1|t+1) = \hat{\alpha}(t+1|t) + L(t+1)h\left[\bar{y}(t), \psi^{-1}(\hat{\alpha}(t+1|t)\right] \\ P(t+1|t+1) = \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + L(t+1)D_+(t+1)R_n(t+1)D_+^T(t+1)L^T(t+1) \end{cases} \tag{F.28}$$

where now the quantities $L(t+1), \Lambda(t+1)$ and $\Gamma(t+1)$ are defined according to section F.5. Note that we have reduced the size of the state from $n+m$ down to m.

F.5 Extended Kalman Filtering for implicit measurement constraints

We are interested in building an estimator for a process $\{\alpha\}$ which is described by a stochastic difference equation of the form

$$\alpha(t+1) = f(\alpha(t)) + v(t) \ ; \ \alpha(t_0) = \alpha_0$$

where $v(t) \in \mathcal{N}(0, Q_v)$ is a white, zero-mean Gaussian noise with variance Q_v. Suppose there is a measurable quantity $x(t)$ which is linked to α by the constraint

$$h[\alpha(t), x(t)] = 0 \ \forall t. \tag{F.29}$$

We will assume throughout $f, h \in C^r \ ; \ r \geq 1$. Usually x is known via some noisy measurement:

$$x(t) = y(t) + w(t) \ : \ w(t) \in \mathcal{N}(0, R_w) \tag{F.30}$$

where the variance/covariance matrix R_w is derived from knowledge of the measurement device. The model we consider is hence of the form

$$\begin{cases} \alpha(t+1) = f(\alpha(t)) + v(t) \ ; \ \alpha(t_0) = \alpha_0 \\ h[\alpha(t), y(t) + w(t)] = 0 \end{cases} . \tag{F.31}$$

Construction of the variational model about the reference trajectory

Consider at each time sample t a reference trajectory $\bar{\alpha}(t)$ which solves the difference equation

$$\bar{\alpha}(t+1) = f(\bar{\alpha}(t))$$

and the jacobian matrix

$$F(\bar{\alpha}(t)) \doteq F(t) = \left(\frac{\partial f}{\partial \alpha}\right)_{|_{\bar{\alpha}(t)}} .$$

The linearization of the measurement equation about the point $(\bar{\alpha}(t), y(t))$ is

$$h[\alpha(t), x(t)] = h[\bar{\alpha}(t), y(t)] + C(\bar{\alpha}, y)(\alpha(t) - \bar{\alpha}(t)) + D(\bar{\alpha}, y)(x(t) - y(t)) + \mathcal{O}(\mathcal{E}^2)$$

where

$$\begin{aligned} C(\bar{\alpha}, y) &\doteq \left(\frac{\partial h}{\partial \alpha}\right)_{|_{\bar{\alpha}(t), y(t)}} \\ D(\bar{\alpha}, y) &\doteq \left(\frac{\partial h}{\partial x}\right)_{|_{\bar{\alpha}(t), y(t)}} \\ \mathcal{E}^2 &\doteq \{\|\alpha - \bar{\alpha}\|^2, \|x - y\|^2\} \end{aligned}$$

and the limit implicit in $\mathcal{O}$ is intended in the mean-square sense. Exploiting the fact that $h[\alpha, x] = 0$, calling $\delta\alpha(t) \doteq \alpha(t) - \bar{\alpha}(t)$ and neglecting the arguments in C and D, we have, up to second-order terms

$$h[\bar{\alpha}(t), y(t)] = -C\delta\alpha(t) - Dw(t).$$

Prediction Step

Suppose at some time t we have available the best estimate $\hat{\alpha}(t|t)$; we may write the variational model about the trajectory $\bar{\alpha}(t)$ defined such that

$$\bar{\alpha}(t+1) = f(\bar{\alpha}(t)) \; ; \; \bar{\alpha}(t) = \hat{\alpha}(t|t).$$

For small displacements we may write

$$\delta\alpha(t+1) = F(\bar{\alpha}(t))\delta\alpha(t) + \tilde{v}(t) \tag{F.32}$$

where the noise term $\tilde{v}(t)$ may include a linearization error component.

Note that with such a choice we have $\delta\hat{\alpha}(t|t) = 0$ and $\delta\hat{\alpha}(t+1|t) = F(\bar{\alpha}(t))\delta\hat{\alpha}(t|t) = 0$, from which we can conclude

$$\hat{\alpha}(t+1|t) = \bar{\alpha}(t+1) = f(\bar{\alpha}(t)) = f(\hat{\alpha}(t|t)). \tag{F.33}$$

The variance of the prediction error $\delta\hat{\alpha}(t+1|t)$ is

$$P(t+1|t) = F(t)P(t|t)F^T(t) + \tilde{Q} \tag{F.34}$$

where $\tilde{Q} = var(\tilde{v})$. The last two equations represent the prediction step for the estimator and are equal, as expected, to the prediction of the explicit EKF [58, 55, 17].

Update Step

At time $t+1$ a new measurement becomes available $y(t+1)$, which is used to update the prediction $\hat{\alpha}(t+1|t)$ and its error variance $P(t+1|t)$. Exploiting the linearization of the measurement equation about $\bar{\alpha}(t+1) = \hat{\alpha}(t+1|t)$, we obtain, letting $\hat{\alpha} \doteq \hat{\alpha}(t+1|t)$ and $y \doteq y(t+1)$,

$$h[\hat{\alpha}, y] = -C(\hat{\alpha}, y)\delta\alpha(t+1) - n(t+1) \tag{F.35}$$

where we have defined $n(t+1) \doteq D(\hat{\alpha}, y)w(t+1)$. This, together with the equation (F.32) defines a linear and *explicit* variational model, for which we can finally write the update equation based on the traditional linear Kalman filter:

$$\delta\hat{\alpha}(t+1|t+1) = \delta\hat{\alpha}(t+1|t) + L(t+1)\left[h[\hat{\alpha}, y] + C(\hat{\alpha}, y)\delta\hat{\alpha}(t+1|t)\right] \tag{F.36}$$

where

$$L(t+1) = -P(t+1|t)C(\hat{\alpha}, y)^T\Lambda^{-1}(t+1) \tag{F.37}$$

$$\Lambda(t+1) = C(\hat{\alpha}, y)P(t|t)C(\hat{\alpha}, y)^T + R_n(t+1) \tag{F.38}$$

$$P(t+1|t+1) = \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + L(t+1)R_n(t+1)L(t+1)^T \tag{F.39}$$

$$\Gamma(t+1) = (I - L(t+1)C(\hat{\alpha}, y)). \tag{F.40}$$

Since $\delta\hat{\alpha}(t+1|t) = 0$ and $\delta\hat{\alpha}(t+1|t+1) = \hat{\alpha}(t+1|t+1) - \hat{\alpha}(t+1|t)$, we may write the update equation for the original model:

$$\hat{\alpha}(t+1|t+1) = \hat{\alpha}(t+1|t) + L(t+1)h\left[\hat{\alpha}(t+1|t), y(t+1)\right]. \tag{F.41}$$

In this formulation the quantity $h\left[\hat{\alpha}(t+1|t), y(t+1)\right]$ plays the role of the pseudo-innovation. The noise n defined in (F.35) has a variance which is calculated from its definition:

$$R_n(t) = D(\hat{\alpha}, y)R_w(t)D^T(\hat{\alpha}, y). \tag{F.42}$$

The update of the variance $P(t+1|t+1)$ is computed from the standard equations of the linear Kalman filter [23, 33, 70, 110]. See also [33, 70, 110, 46].